

КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ

ИНСТИТУТ ФИЗИКИ

Кафедра радиоастрономии

Ю.М.Стенин, В.Ю.Теплов

АРХИТЕКТУРА МИКРОКОНТРОЛЛЕРА C8051 (SILABS)

Учебно-методическое пособие

Казань 2012

УДК 681.322

Принято на заседании кафедры радиоастрономии
Протокол №7 от 22 марта 2012 года

Рецензент:

Доцент кафедры радиофизики КФУ, кандидат физико-математических наук Латыпов Р. Р.

Ю.М.Стенин, В.Ю.Теплов. АРХИТЕКТУРА МИКРОКОНТРОЛЛЕРА C8051 (SILABS): учебно-методическое пособие/ Ю.М. Стенин, В.Ю. Теплов. - Казань: КФУ, 2012. 48 с.

Данная работа предназначена для использования при выполнении лабораторного практикума по дисциплине «Микропроцессоры и автоматизация эксперимента» студентами 3 курса кафедры радиоастрономии Института физики КФУ. Содержит описание архитектуры микроконтроллеров семейства Intel 8051 и конкретно МК C8051F064 Silicon Laboratories, введение в интегрированную среду разработки программ IDE SiLabs, пояснения к разработке программ, задания для самостоятельной работы и контрольные вопросы по изучаемой теме.

© Ю.М.Стенин, **В.Ю.Теплов**
© Казанский (Приволжский)
Федеральный университет, 2012

ОГЛАВЛЕНИЕ

1. ЦЕЛЬ И СОДЕРЖАНИЕ РАБОТЫ.....	5
2. ЗАДАНИЯ.....	5
2.1. Изучение особенностей однокристальных микроконтроллеров.....	5
2.2. Изучение архитектуры микроконтроллеров семейства Intel 8051	5
2.3. Изучение особенностей архитектуры микроконтроллеров C8051F064 SiLabs	5
2.4. Знакомство с системами и средой программирования микроконтроллеров	5
2.5. Изучение текстов прилагаемой программы на языке ассемблера и языке C....	5
3. АРХИТЕКТУРА МИКРОКОНТРОЛЛЕРА INTEL 8051.....	6
3.1. Арифметико-логическое устройство (АЛУ, ALU)	6
3.2. Резидентная память программ и данных	7
3.3. Аккумулятор, регистры общего назначения, регистр PSW	8
3.4. Регистры-указатели.....	10
3.5. Регистры специальных функций	10
3.6. Устройство управления и синхронизации.....	10
3.7. Параллельные порты ввода/вывода информации	11
3.8. Таймер/счётчик.....	12
3.9. Последовательный порт	15
3.9.1. Регистр SBUF	15
3.9.2. Регистр SCON	15
3.9.3. Работа UART в мультимикроконтроллерных системах	17
3.9.4. Скорость приёма/передачи	17
3.10. Система прерываний.....	18
4. ОСОБЕННОСТИ МК C8051F064 SILABS.....	20
4.1. Процессорное ядро CIP-51	22
4.1.1. Полностью 8051-совместимая архитектура.....	22
4.1.2. Улучшенная производительность.....	22
4.1.3. Дополнительные функции.....	23
4.1.4. Встроенная память	24
4.1.5. JTAG отладчик и интерфейс граничного сканирования	26
4.1.6. Программируемые порты ввода/вывода и матрица соединений.....	27
4.1.7. Программируемый массив счетчиков (ПМС)	28
4.1.8. Последовательные порты	29
4.1.9. 16-разрядный аналого-цифровой преобразователь	30
4.1.10. Аналоговые компараторы.....	31
5. ЛАБОРАТОРНАЯ УСТАНОВКА C8051F064EK EVALUATION KIT.....	31
5.1. Плата начального освоения C8051F064EK	32
5.2. Источник системной опорной частоты	33

5.3. Переключатели и светодиоды	33
6. ВВЕДЕНИЕ В IDE SILABS.....	34
6.1. Оптимизирующий кросс-компилятор C51 (Keil C51)	34
6.2. Макроассемблер A51 (Keil Software A51).....	35
6.3. Компоновщик L51 (BL51)	35
6.4. Отладчик/симулятор.....	35
7. ЗНАКОМСТВО С РАБОТОЙ 8051 F064ЕК	36
7.1. Запуск IDE Silabs и открытие файла проекта	37
7.2. Запуск IDE Silabs и создание файла проекта с использованием компилятора языка Си.....	41
7.3. Работа с отладчиком C51	44
8. КОНТРОЛЬНЫЕ ВОПРОСЫ.....	46
9. ЗАДАНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ	47
10. СОДЕРЖАНИЕ ОТЧЁТА	47
СПИСОК ЛИТЕРАТУРЫ	48

1. ЦЕЛЬ И СОДЕРЖАНИЕ РАБОТЫ

Целью работы является изучение архитектуры восьмиразрядных микроконтроллеров семейства Intel 8051 и его расширенного варианта - микроконтроллера C8051F064 компании Silicon Laboratories (SiLabs), а также освоение интегрированной среды IDE SiLabs разработки программного обеспечения для этого МК.

В процессе подготовки к работе изучается структура базового микроконтроллера Intel 8051, его функциональные узлы и особенности их работы, а также особенности организации МК C8051F064.

На конкретном примере изучаются этапы технологии разработки и отладки программ, основные приёмы работы с IDE SiLabs. После выполнения работы оформляется отчёт.

За основу пособия была взята работа [1].

2. ЗАДАНИЯ

2.1. Изучение особенностей однокристальных микроконтроллеров

Семейства однокристальных микроконтроллеров. Функциональные возможности и области применения.

2.2. Изучение архитектуры микроконтроллеров семейства Intel 8051

Восьмиразрядные микроконтроллеры 51-го семейства (Intel 8051, i8051, x51). Арифметико-логическое устройство, память данных и память команд, регистры общего назначения и регистры специальных функций, устройство управления и синхронизации, таймеры-счётчики, параллельные порты ввода-вывода, последовательный порт. Система прерываний. Расширение памяти программ и памяти данных [2 - 6].

2.3. Изучение особенностей архитектуры микроконтроллеров C8051F064 SiLabs

Микроконтроллеры 52-ого семейства (Intel8052, i8052, x52). Структура памяти, массив таймеров, аналоговый компаратор, 16-разрядные АЦП, конфигурирование портов ввода/вывода, последовательные порты, JTAG-отладчик [7 – 11].

2.4. Знакомство с системами и средой программирования микроконтроллеров

Кросс-компилятор C51. Макроассемблер A51. компоновщик L51. Отладчик/симулятор. Запуск IDE Silabs и открытие файла проекта, использующего язык ассемблера. Создание файла проекта с использованием компилятора C.

2.5. Изучение текстов прилагаемой программы на языке ассемблера и языке C.

3. АРХИТЕКТУРА МИКРОКОНТРОЛЛЕРА INTEL 8051

В микропроцессорной технике выделился самостоятельный класс интегральных схем – микроконтроллеры, которые предназначены для встраивания в приборы различного назначения. От класса микропроцессоров их отличает наличие встроенной памяти, развитые средства взаимодействия с внешними устройствами.

Микроконтроллер i8051 выполнен на основе высокоуровневой n-МОП технологии. Через четыре программируемых параллельных порта ввода/вывода и один последовательный порт микроконтроллер взаимодействует с внешними устройствами. Основу структурной схемы МК (рис. 1) образует внутренняя двунаправленная 8-битная шина, которая связывает между собой основные узлы и устройства микроконтроллера: резидентную память программ (RPM), резидентную память данных (RDM), арифметико-логическое устройство (ALU), блок регистров специальных функций, устройство управления (CU) и порты ввода/вывода (P0 – P3).

3.1. Арифметико-логическое устройство (АЛУ, ALU)

Восьмиразрядное АЛУ предназначено для выполнения арифметических операций сложения, вычитания, умножения и деления; логических операций «И», «ИЛИ», «исключающее ИЛИ», а также операций циклического сдвига, сброса, инвертирования и т.п. К входам АЛУ подключены программно-недоступные регистры T1 и T2, предназначенные для временного хранения операндов, блок десятичной коррекции (DCU) и блок формирования признаков результата операции (PSW).

Простейшая операция сложения используется в ALU для инкрементирования содержимого регистров, продвижения регистра-указателя данных (RAR) и автоматического вычисления следующего адреса резидентной памяти программ. Простейшая операция вычитания используется в ALU для декрементирования регистров и сравнения переменных.

Простейшие операции автоматически образуют “танделы” для выполнения таких операций, как, например, инкрементирование 16-битных регистровых пар. В ALU реализуется механизм каскадного выполнения простейших операций для реализации сложных команд. Так, например, при выполнении одной из команд условной передачи управления, по результату сравнения в ALU трижды инкрементируется счётчик команд (PC), дважды производится чтение из RDM, выполняется арифметическое сравнение двух переменных, формируется 16-битный адрес перехода и принимается решение о том, делать или не делать переход по программе. Все перечисленные операции выполняются всего лишь за 2 мкс.

Важной особенностью ALU является его способность оперировать не только байтами, но и битами. Отдельные программно-доступные биты могут быть установлены, сброшены, инвертированы, переданы, проверены и использованы в логических операциях. Эта способность достаточно важна, поскольку для управления объектами часто применяются алгоритмы, содержащие операции над входными и выходными булевыми переменными, реализация которых средствами обычных микропроцессоров сопряжена с определенными трудностями.

Память данных RDM предназначена для хранения переменных в процессе выполнения программы, адресуется одним байтом и имеет емкость 128 байт. Кроме того, к её адресному пространству примыкают адреса регистров специальных функций, которые перечислены в табл. 1.

Память программ и данных можно расширить до 64 Кбайт путем подключения внешних микросхем.

Таблица 1

Блок регистров специальных функций

Символ	Наименование	Адрес
* A	Аккумулятор	0E0H
* B	Регистр-расширитель аккумулятора	0F0H
* PSW	Слово состояния программы	0D0H
SP	Регистр-указатель стека	81H
DPTR	Регистр-указатель данных (DPH)	83H
	(DPL)	82H
* P0	Порт 0	80H
* P1	Порт 1	90H
* P2	Порт 2	0A0H
* P3	Порт 3	0B0H
* IP	Регистр приоритетов прерываний	0B8H
* IE	Регистр маски прерываний	0A8H
TMOD	Регистр режима таймера/счётчика	89H
* TCON	Регистр управления/статуса таймера	88H
TH0	Таймер 0 (старший байт)	8CH
TL0	Таймер 0 (младший байт)	8AH
TH1	Таймер 1 (старший байт)	8DH
TL1	Таймер 1 (младший байт)	8BH
* SCON	Регистр управления приёмопередатчиком	98H
SBUF	Буфер приёмопередатчика	99H
PCON	Регистр управления мощностью	87H

Примечание. Регистры со значком () допускают адресацию отдельных битов*

3.3. Аккумулятор, регистры общего назначения, регистр PSW

Регистр-аккумулятор (A) является источником операнда и местом фиксации результата при выполнении арифметических, логических операций и ряда операций передачи данных. Кроме того, только с использованием аккумулятора могут быть выполнены операции сдвигов, проверка на нуль, формирование флага паритета и т.п. В распоряжении пользователя имеются 8 регистров общего назначения (РОН) R0–R7 одного из четырёх возможных банков. При выполнении многих команд в ALU формируется ряд признаков операции (флагов), которые фиксируются в регистре PSW. В табл. 2 приводится перечень флагов PSW, даются их символические имена и описываются условия их формирования.

Таблица 2

Формат слова состояния программы PSW

Символ	Разряд	Имя и назначение
C	PSW.7	Флаг переноса. Устанавливается и сбрасывается аппаратно или программно при выполнении арифметических и логических операций
AC	PSW.6	Флаг вспомогательного переноса. Устанавливается и сбрасывается только аппаратно при выполнении команд сложения и вычитания и сигнализирует о переносе или займе в бите 3
F0	PSW.5	Флаг 0. Может быть установлен, сброшен или проверен программой как флаг, специфицируемый пользователем
RS1	PSW.4	Выбор банка регистров. Устанавливается и сбрасывается программно
RS0	PSW.3	для выбора рабочего банка регистров (табл. 3)
OV	PSW.2	Флаг переполнения. Устанавливается и сбрасывается аппаратно при выполнении арифметических операций
-	PSW.1	Не используется
P	PSW.0	Флаг паритета. Устанавливается и сбрасывается аппаратно в каждом цикле и фиксирует нечётное/чётное число единичных битов в аккумуляторе, т.е. выполняет контроль по четности

Таблица 3

Выбор рабочего банка регистров

RS1	RS0	Банк	Границы адресов
0	0	0	00H – 07H
0	1	1	08H – 0FH
1	0	2	10H – 17H
1	1	3	18H – 1FH

Наиболее “активным” флагом PSW является флаг переноса C, который принимает участие и модифицируется в процессе выполнения множества операций, включая сложение, вычитание и сдвиги. Кроме того, флаг переноса выполняет функции «булева» аккумулятора в командах, манипулирующих с битами. Флаг переполнения OV фиксирует арифметическое переполнение при операциях над целыми числами со знаком и делает возможным использование арифметики в дополнительных кодах. Значение флагов RS0, RS1 выбора банка PОН полностью определяется выполняемой программой и используется для выбора одного из четырёх регистровых банков.

В микропроцессорах большинство команд выполняется, используя регистр-аккумулятор A. В отличие от этого, микроконтроллер может выполнять множество команд и без его участия. Так, данные могут быть переданы из любой ячейки RDM в любой регистр, любой регистр может быть загружен непосредственным операндом; переменные могут быть инкрементированы, декрементированы и проверены без использования аккумулятора, как и выполнены многие логические операции. Флаги и управляющие биты могут быть проверены и изменены аналогично.

3.4. Регистры-указатели

Восьмиразрядный указатель стека SP может адресовать любую область RDM. Его содержимое инкрементируется прежде, чем данные будут запомнены в стеке в ходе выполнения команд PUSH и CALL. Содержимое SP декрементируется после выполнения команд POP и RET. Подобный способ адресации элементов стека называют преинкрементным/постдекрементным. В процессе инициализации микроконтроллера после сигнала RST в SP автоматически загружается код 07H. Это значит, что если прикладная программа не переопределяет стек, то первый элемент данных в стеке будет располагаться в ячейке RDM с адресом 08H.

Двухбайтный регистр-указатель данных DPTR обычно используется для фиксации 16-битного адреса в операциях с обращением к внешней памяти. Командами микроконтроллера DPTR может быть использован или как 16-битный регистр, или как два независимых 8-битных регистра (DPH и DPL).

3.5. Регистры специальных функций

Регистры с именами IP, IE, TMOD, TCON, SCON и PCON используются для фиксации и программного изменения управляющих бит и бит состояния схемы прерывания, таймера/счётчика, приёмопередатчика последовательного порта и для управления энергопотреблением. Их организация будет описана ниже при рассмотрении особенностей работы микроконтроллера в различных режимах.

3.6. Устройство управления и синхронизации

Кварцевый резонатор, подключаемый к внешним выводам микроконтроллера, управляет работой внутреннего генератора, который в свою очередь формирует сигналы синхронизации. Устройство управления CU на основе сигналов синхронизации формирует машинный цикл фиксированной длительности, равной 12 периодам резонатора. Большинство команд микроконтроллера выполняется за один машинный цикл. Некоторые команды, оперирующие с 2-байтными словами или связанные с обращением к внешней памяти, выполняются за два машинных цикла. Только команды деления и умножения требуют четырех машинных циклов. На основе этих особенностей работы устройства управления производится расчёт времени исполнения прикладных программ.

На схеме микроконтроллера к устройству управления примыкает регистр команд IR. В его функцию входит хранение кода выполняемой команды.

Входные и выходные сигналы устройства управления и синхронизации:

- PSEN – разрешение программной памяти,
- ALE – выходной сигнал разрешения фиксации адреса,
- PROG – сигнал программирования,
- EA – блокировка работы с внутренней памятью,
- VPP – напряжение программирования,
- RST – сигнал общего сброса,
- VPD – вывод резервного питания памяти от внешнего источника,
- XTAL – входы подключения кварцевого резонатора.

3.7. Параллельные порты ввода/вывода информации

Все четыре порта (P0-P3) предназначены для ввода или вывода информации побайтно. Каждый порт содержит регистр-защёлку, входной буфер и выходной драйвер.

Выходные драйверы портов 0 и 2, а также входной буфер порта 0 используются при обращении к внешней памяти. При этом через порт 0 в режиме временно-го мультиплексирования сначала выводится младший байт адреса, а затем выдается или принимается байт данных. Через порт 2 выводится старший байт адреса в тех случаях, когда разрядность адреса равна 16 бит.

Все выходы порта 3 могут быть использованы для реализации альтернативных функций, перечисленных в табл. 4. Эти функции могут быть задействованы путем записи 1 в соответствующие биты регистра-защёлки (P3.0-P3.7) порта 3.

Таблица 4

Альтернативные функции порта P3

Символ	Разряд	Имя и назначение
RD	P3.7	Чтение. Активный сигнал низкого уровня формируется аппаратно при обращении к внешней памяти данных
WR	P3.6	Запись. Активный сигнал низкого уровня формируется аппаратно при обращении к внешней памяти данных
T1	P3.5	Вход таймера/счётчика 1 или тест-вход
T0	P3.4	Вход таймера/счётчика 0 или тест-вход
INT1	P3.3	Вход запроса прерывания 1. Воспринимается сигнал низкого уровня или срез
INT0	P3.2	Вход запроса прерывания 0. Воспринимается сигнал низкого уровня или срез
TXD	P3.1	Выход передатчика последовательного порта в режиме UART. Выход синхронизации в режиме регистра сдвига
RXD	P3.0	Вход приёмника последовательного порта в режиме UART. Ввод/вывод данных в режиме регистра сдвига

Порт 0 является двунаправленным, а порты 1–3 – квазидвунаправленными. Каждая линия портов может быть использована независимо для ввода или вывода.

По сигналу RST в регистры-защёлки всех портов автоматически записываются единицы, настраивающие их тем самым на режим ввода.

Все порты могут быть использованы для организации ввода/вывода информации по двунаправленным линиям передачи. Однако порты 0 и 2 не могут быть использованы для этой цели в случае, если система имеет внешнюю память, связь с которой организуется через общую разделяемую шину адреса/данных, работающую в режиме временного мультиплексирования.

Обращение к портам ввода/вывода возможно с использованием команд, оперирующих с байтом, отдельным битом, произвольной комбинацией битов. При этом в тех случаях, когда порт является одновременно операндом и местом назначения результата, устройство управления автоматически реализует специальный режим, который называется “чтение-модификация-запись”. Этот режим обращения

предполагает ввод сигналов не с внешних выводов порта, а из его регистра-защёлки, что позволяет исключить неправильное считывание ранее выведенной информации. Этот механизм обращения к портам реализован в командах:

- ANL – логическое «И», например ANL P1,A;
- ORL – логическое «ИЛИ», например ORL P2,A;
- XRL – «исключающее ИЛИ», например XRL P3,A;
- JBC – переход, если в адресуемом бите единица, и последующий сброс бита, например JBC P1.1, LABEL;
- CPL – инверсия бита, например CPL P3.3;
- INC – инкремент порта, например INC P2;
- DEC – декремент порта, например DEC P2;
- DJNZ – декремент порта и переход, если его содержимое не равно нулю, например DJNZ r, LABEL;
- MOV PX.Y,C – передача бита переноса в бит Y порта X;
- SET PX.Y – установка бита Y порта X;
- CLR PX.Y – сброс бита Y порта X.

3.8. Таймер/счётчик

В составе микроконтроллера имеются регистровые пары с символическими именами TH0, TL0 и TH1, TL1, на основе которых функционируют два независимых программно-управляемых 16-битных таймера/счётчика событий (T/C0 и T/C1). При работе в качестве таймера содержимое T/C инкрементируется в каждом машинном цикле, то есть через каждые 12 периодов резонатора. При работе в качестве счётчика содержимое T/C инкрементируется под воздействием перехода из 1 в 0 внешнего входного сигнала, подаваемого на соответствующий (T0, T1) вход микроконтроллера. Опрос сигналов выполняется в каждом машинном цикле. Так как на распознавание перехода требуются два машинных цикла, то максимальная частота подсчёта входных сигналов равна 1/24 частоты резонатора. На длительность периода входных сигналов ограничений сверху нет. Для гарантированного прочтения входного считываемого сигнала он должен удерживать значение 1 как минимум в течение одного машинного цикла.

Для управления режимами работы и для организации взаимодействия таймеров с системой прерывания используются два регистра специальных функций TMOD и TCON, описание которых приводится в табл. 5-7. Для обоих T/C режимы работы 0, 1 и 2 одинаковы. Режимы 3 для T/C0 и T/C1 различны.

Регистр режима работы таймера/счётчика

Символ	Разряд	Имя и назначение
GATE	TMOD.7 для T/C1 TMOD.3 для T/C0	Управление блокировкой. Если бит установлен, то таймер/счётчик “х” разрешен до тех пор, пока на входе “INT х” высокий уровень и бит управления “TRx” установлен. Если бит сброшен, то T/C разрешается, как только бит управления “TRx” устанавливается
C/T	TMOD.6 для T/C1 TMOD.2 для T/C0	Бит выбора режима таймера или счётчика событий. Если бит сброшен, то работает таймер от внутреннего источника сигналов синхронизации. Если бит установлен, то работает счётчик от внешних сигналов на входе “Tx”
M1	TMOD.5 для T/C1 TMOD.1 для T/C0	Режим работы (см. табл. 6)
M0	TMOD.4 для T/C1 TMOD.0 для T/C0	

Таблица 6

Режимы работы таймера/счётчика

M1	M0	Режим работы
0	0	“TLx” работает как 5-битный предделитель
0	1	16-битный таймер/счётчик. “TNx” и “TLx” включены последовательно
1	0	8-битный автоперезагружаемый таймер/счётчик. “TNx” хранит значение, которое должно быть перезагружено в “TLx” каждый раз по переполнению
1	1	Таймер/счётчик 1 останавливается. Таймер/счётчик 0: TL0 работает как 8-битный таймер/счётчик, и его режим определяется управляющими битами таймера 0. TN0 работает только как 8-битный таймер, и его режим определяется управляющими битами таймера 1

Режим 0. Перевод любого T/C в этот режим делает его 8-разрядным таймером, на вход которого подключен 5-битный предделитель частоты на 32. В этом режиме таймерный регистр имеет разрядность 13 бит. При переходе из состояния “все единицы” в состояние “все нули” устанавливается флаг прерывания от таймера TF1. Входной синхросигнал таймера 1 разрешен (поступает на вход T/C), когда управляющий бит TR1 установлен в 1 и либо управляющий бит GATE (блокировка) равен 0, либо на внешний вход запроса прерывания INT1 поступает уровень 1.

Установка бита GATE в 1 позволяет использовать таймер для измерения длительности импульсного сигнала, подаваемого на вход запроса прерывания.

Таблица 7

Регистр управления/статуса таймера

Символ	Разряд	Имя и назначение
TF1	TCON.7	Флаг переполнения таймера 1. Устанавливается аппаратно при переполнении таймера/счётчика. Сбрасывается при обслуживании прерывания аппаратно
TR1	TCON.6	Бит управления таймера 1. Устанавливается/сбрасывается программой для пуска/останова
TF0	TCON.5	Флаг переполнения таймера 0. Устанавливается аппаратно. Сбрасывается при обслуживании прерывания
TR0	TCON.4	Бит управления таймера 0. Устанавливается/сбрасывается программой для пуска/останова таймера/счётчика
IE1	TCON.3	Флаг фронта прерывания 1. Устанавливается аппаратно, когда детектируется срез внешнего сигнала INT1. Сбрасывается при обслуживании прерывания
IT1	TCON.2	Бит управления типом прерывания 1. Устанавливается/сбрасывается программно для спецификации запроса INT1 (срез/низкий уровень)
IE0	TCON.1	Флаг фронта прерывания 0. Устанавливается по срезу сигнала INT0. Сбрасывается при обслуживании прерывания
IT0	TCON.0	Бит управления типом прерывания 0. Устанавливается/сбрасывается программно для спецификации запроса INT0 (срез/низкий уровень)

Режим 1. Работа любого Т/С в этом режиме такая же, как и в режиме 0, за исключением того, что таймерный регистр имеет разрядность 16 бит.

Режим 2. В этом режиме работа организована таким образом, что переполнение (переход из состояния “все единицы” в состояние “все нули”) 8-битного счётчика TL1 приводит не только к установке флага TF1, но и автоматически перезагружает в TL1 содержимое старшего байта (TH1) таймерного регистра, которое предварительно было задано программным путем. Перезагрузка оставляет содержимое TH1 неизменным. В режиме 2 Т/С0 и Т/С1 работают совершенно одинаково.

Режим 3. В этом режиме Т/С0 и Т/С1 работают по-разному. Т/С1 сохраняет неизменным своё текущее содержимое. Иными словами, эффект такой же, как и при сбросе управляющего бита TR1 в нуль. В этом режиме TL0 и TH0 функционируют как два независимых 8-битных счётчика. Работу TL0 определяют управляющие биты Т/С0 (С/Т, GATE, TR0), входной сигнал INT0 и флаг переполнения TF0. Работу TH0, который может выполнять только функции таймера (подсчёт машинных циклов микроконтроллера), определяет управляющий бит TR1. При этом TH0 использует флаг переполнения TF1.

Режим 3 используется в тех случаях, когда требуется наличие дополнительного 8-битного таймера или счётчика событий. Можно считать, что в режиме 3 микроконтроллер имеет в своем составе три таймера/счётчика. В том случае, если Т/С0 используется в режиме 3, Т/С1 может быть или включен, или выключен, или переведен в свой собственный режим 3, или может быть использован последовательным портом в качестве генератора частоты передачи, или, наконец, может быть использован в любом применении, не требующем прерывания.

3.9. Последовательный порт

Через универсальный асинхронный приёмопередатчик UART (Universal Asynchronous Receiver-Transmitter) происходит передача информации, представленной последовательным кодом (младшими битами вперед), в полнодуплексном режиме обмена. В состав UART, называемого часто последовательным портом, входят принимающий и передающий сдвигающие регистры, а также специальный буферный регистр (SBUF) приёмопередатчика.

3.9.1. Регистр SBUF

Представляет собой два независимых регистра: буфер приёмника и буфер передатчика. Загрузка байта в SBUF немедленно вызывает начало процесса передачи через последовательный порт. Когда байт считывается из SBUF, это значит, что его источником является приёмник последовательного порта. Запись байта в буфер приводит к автоматической переписи байта в сдвигающий регистр передатчика и инициирует начало передачи байта. Наличие буферного регистра приёмника позволяет совмещать операцию чтения ранее принятого байта с приёмом очередного байта. Если к моменту окончания приёма байта предыдущий байт не был считан, то он будет потерян.

Последовательный порт может работать в четырех различных режимах.

Режим 0. Информация передаётся и принимается через вход приёмника RXD. Принимаются и передаются 8 бит данных. Через внешний выход передатчика TXD выдаются импульсы сдвига, которые сопровождают каждый бит. Частота передачи равна $1/12$ частоты резонатора.

Режим 1. Через TXD передаются или из RXD принимаются 10 бит: старт-бит (0), 8 бит данных и стоп-бит (1). Скорость приёма/передачи – величина переменная и задаётся таймером.

Режим 2. Через TXD передаются или из RXD принимаются 11 бит: старт-бит, 8 бит данных, программируемый девятый бит и стоп-бит. При передаче девятый бит может использоваться для повышения достоверности передачи путём контроля по чётности и в него можно поместить значение признака паритета из PSW. Частота приёма/передачи выбирается программно и может быть равна $1/32$ или $1/64$ частоты резонатора в зависимости от SMOD.

Режим 3. Совпадает с режимом 2, но частота приёма/передачи является величиной переменной и задаётся таймером.

3.9.2. Регистр SCON

Регистр предназначен для управления режимом работы UART. Регистр содержит управляющие биты и девятый бит принимаемых или передаваемых данных RB8 и TB8, а также биты прерывания приёмопередатчика RI и TI. Функциональное назначение битов указано в табл. 8 и 9.

Таблица 8

Регистр управления/статуса UART

Символ	Разряд	Имя и назначение
SM0	SCON.7	Биты управления режимом работы UART. Устанавливаются/сбрасываются программно (табл. 9)
SM1	SCON.6	
SM2	SCON.5	Бит управления режимом UART. Устанавливается программно для запрета приёма сообщения, в котором девятый бит равен 0
REN	SCON.4	Бит разрешения приёма. Устанавливается/сбрасывается программно для разрешения/запрета приёма последовательных данных
TB8	SCON.3	Передача бита 8. Устанавливается/сбрасывается программно для задания девятого передаваемого бита в режиме UART - 9 бит
RB8	SCON.2	Приём бита 8. Устанавливается/сбрасывается аппаратно для фиксации девятого принимаемого бита в режиме UART - 9 бит
TI	SCON.1	Флаг прерывания передатчика. Устанавливается аппаратно при окончании передачи байта. Сбрасывается программно после обслуживания прерывания
RI	SCON.0	Флаг прерывания приёмника. Устанавливается аппаратно при приёме байта. Сбрасывается программно после обслуживания прерывания

Таблица 9

Режим работы UART

SM0	SM1	Режим работы UART
0	0	Сдвигающий регистр расширения ввода/вывода
0	1	UART - 8 бит. Изменяемая скорость передачи
1	0	UART - 9 бит. Фиксированная скорость передачи
1	1	UART - 9 бит. Изменяемая скорость передачи

Режим работы UART задаётся выполняемой программой путём загрузки в два старших разряда SCON. Во всех режимах передача инициируется любой командой, где SBUF указан как получатель байта. Приём в UART в режиме 0 происходит при условии RI=0 и REN=1. В режимах 1-3 приём начинается с приходом стартового бита, если REN=1.

В TB8 программно устанавливается значение девятого бита данных, который будет передан в режиме 2 или 3. В RB8 фиксируется в режимах 2 и 3 девятый принимаемый бит данных. В режиме 1, если SM2=0, в бит RB8 заносится стоп-бит. В режиме 0 RB8 не используется.

Флаг прерывания передатчика TI устанавливается аппаратно в конце периода передачи восьмого бита данных в режиме 0 и в начале периода передачи стоп-бита в режимах 1-3. Подпрограмма обслуживания этого прерывания должна сбрасывать бит TI.

Флаг прерывания приёмника RI устанавливается аппаратно в конце периода приёма восьмого бита данных в режиме 0 и в середине периода приёма стоп-бита в режимах 1-3. Подпрограмма обслуживания прерывания должна сбрасывать бит RI.

3.9.3. Работа UART в мультиконтроллерных системах

В системах управления, которые используются в топологически распределённых объектах, возникает задача обмена информацией между множеством микроконтроллеров, объединённых в локальную вычислительно-управляющую сеть. Как правило, локальные сети на основе Intel 8051 имеют магистральную архитектуру с разделяемым моноканалом (коаксиальный кабель, витая пара, оптоволокно), по которому осуществляется обмен информацией между контроллерами.

Бит SM2 в SCON позволяет простыми средствами реализовать межконтроллерный обмен. Механизм обмена построен на том, что в режимах 2 и 3 программируемый девятый бит данных при приёме фиксируется в бите RB8. UART может быть запрограммирован таким образом, что при получении стоп-бита прерывание от приёмника будет возможно только при условии RB8=1. Ведущий контроллер всем ведомым передаёт широковещательное сообщение с байтом-идентификатором абонента, которое отличается от байтов данных только тем, что в его девятом бите содержится 1. Ведомые по этому признаку вызывают подпрограммы сравнения байта-идентификатора с кодом собственного сетевого адреса. Адресуемый контроллер сбрасывает свой SM2 и готовится к приёму блока данных. Остальные ведомые микроконтроллеры оставляют неизменными свои SM2=1 и передают управление основной программе. При SM2=1 информационные байты в сети прерывания не вызывают.

В режиме 1 автономного микроконтроллера SM2 используется для контроля истинности стоп-бита. В режиме 0 SM2 не используется и должен быть сброшен.

3.9.4. Скорость приёма/передачи

Скорость зависит от режима работы UART. В режиме 0 частота зависит только от резонатора: $f_0 = f_{\text{рез}}/12$. За один машинный цикл передаётся один бит.

В режимах 1-3 скорость зависит от значения управляющего бита SMOD в регистре специальных функций PCON (табл. 10).

В режиме 2 частота передачи $f_2 = (2^{\text{SMOD}}/64)f_{\text{рез}}$.

В режимах 1 и 3 в формировании частоты передачи кроме управляющего бита SMOD принимает участие таймер 1. При этом частота передачи зависит от частоты переполнения (OVT1) и определяется следующим образом: $f_{1,3} = (2^{\text{SMOD}}/32)f_{\text{OVT1}}$. Прерывание от таймера 1 в этом случае должно быть заблокировано. Сам T/C1 может работать и как таймер, и как счётчик событий в любом из трёх режимов. Однако наиболее удобно использовать режим таймера с автоперезагрузкой (старшая тетрада TMOD=0010B). При этом частота передачи определяется выражением $f_{1,3} = (2^{\text{SMOD}}/32)(f_{\text{рез}}/12)(256 - \text{TH1})$. В табл. 11 приводится описание способов настройки T/C1 для получения типовых частот передачи данных через UART.

Таблица 10

Регистр управления мощностью PCON

Символ	Разряд	Наименование и функция
SMOD	PCON.7	Удвоенная скорость передачи. Если бит установлен в 1, то скорость передачи вдвое больше, чем при SMOD=0
-	PCON.6-4	Не используются
GF1	PCON.3	Флаги, специфицируемые пользователем (флаги общего назначения)
GF0	PCON.2	
PD	PCON.1	Бит пониженной мощности. При установке в 1 микроконтроллер переходит в режим пониженного энергопотребления
IDL	PCON.0	Бит холостого хода. Если бит установлен в 1, то микроконтроллер переходит в режим холостого хода

Примечание. При одновременной записи 1 в PD и IDL бит PD имеет преимущество. Сброс PCON выполняется путем загрузки в него кода 0XXX0000.

Таблица 11

Настройка таймера 1 для управления частотой работы UART

Частота приёма/ передачи (BAUD RATE)	Частота резонато- ра, МГц	SMOD	Таймер/счётчик 1			
			С/Т	Режим (MODE)	Перезагружа- емое число	
Режим 0, макс.:	1 МГц	12	X	X	X	
Режим 2, макс.:	375 кГц	12	1	X	X	
Режимы 1,3:	62,5 кГц	12	1	0	2	0FFH
	19,2 кГц	11,059	1	0	2	0FDH
	9,6 кГц	11,059	0	0	2	0FDH
	4,8 кГц	11,059	0	0	2	0FAH
	2,4 кГц	11,059	0	0	2	0F4H
	1,2 кГц	11,059	0	0	2	0E8H
	137,5 Гц	11,059	0	0	2	1DH
	110 Гц	6	0	0	2	72H
	110 Гц	12	0	0	1	0FEEBH

3.10. Система прерываний

Внешние прерывания INT0 и INT1 (рис. 2) могут быть вызваны уровнем или переходом сигнала из 1 в 0 на входах микроконтроллера в зависимости от значений управляющих битов IT0 и IT1 в регистре TCON. От внешних прерываний устанавливаются флаги IE0 и IE1 в регистре TCON, которые инициируют вызов соответствующей подпрограммы обслуживания прерывания. Сброс этих флагов выполняется аппаратно только в том случае, если прерывание было вызвано по переходу (срезу) сигнала. Если же прерывание вызвано уровнем входного сигнала, то сбросом флага IE управляет соответствующая подпрограмма обслуживания прерывания путем воздействия на источник прерывания с целью снятия им запроса.

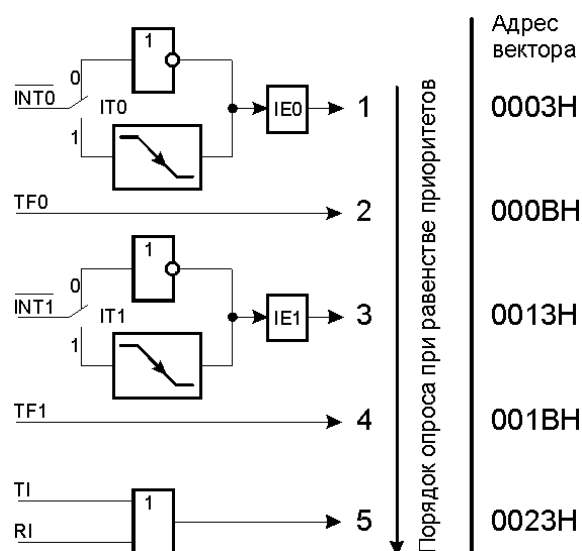


Рис. 2. Схема прерываний

Флаги запросов прерывания от таймеров TF0 и TF1 сбрасываются автоматически при передаче управления подпрограмме обслуживания. Флаги запросов прерывания RI и TI устанавливаются UART аппаратно, но сбрасываться должны программой. Прерывания могут быть вызваны или отменены программой, так как все перечисленные флаги программно доступны.

В блоке регистров специальных функций есть два регистра, предназначенных для управления режимом прерываний и уровнями приоритета. Форматы этих регистров, имеющих символические имена IE и IP, описаны в табл. 12 и 13 соответственно.

Таблица 12

Регистр масок прерывания IE

Символ	Разряд	Имя и назначение
EA	IE.7	Снятие блокировки прерываний. Сбрасывается программно для запрета всех прерываний независимо от состояний IE4-IE0
-	IE.6, 5	Не используются
ES	IE.4	Бит разрешения прерывания от UART. Установка/сброс программой для разрешения/запрета прерываний от флагов TI, RI
ET1	IE.3	Бит разрешения прерывания от таймера 1. Установка/сброс программой для разрешения/запрета прерываний от таймера 1
EX1	IE.2	Бит разрешения внешнего прерывания 1. Установка/сброс программой для разрешения/запрета прерываний
ET0	IE.1	Разрешение прерывания от таймера 0. Работает аналогично IE.3
EX0	IE.0	Разрешения внешнего прерывания 0. Работает аналогично IE.2

Таблица 13

Регистр приоритетов прерывания IP

Символ	Разряд	Имя и назначение
—	IP.7-5	Не используются
PS	IP.4	Бит приоритета UART. Установка/сброс программой для назначения прерыванию от UART высшего/низшего приоритета
PT1	IP.3	Бит приоритета таймера 1. Установка/сброс программой для назначения прерыванию от таймера 1 высшего/низшего приоритета
PX1	IP.2	Бит приоритета внешнего прерывания 1. Установка/сброс программой для назначения прерыванию INT1 высшего/низшего приоритета
PT0	IP.1	Бит приоритета таймера 0. Работает аналогично IP.3
PX0	IP.0	Приоритет внешнего прерывания 0. Работает аналогично IP.2

Возможность программной установки/сброса любого управляющего бита в этих двух регистрах делает систему прерываний исключительно гибкой.

Флаги прерываний опрашиваются в каждом машинном цикле. Ранжирование прерываний по приоритету выполняется в течение следующего машинного цикла. Система прерываний сформирует аппаратно вызов LCALL соответствующей подпрограммы обслуживания, если она не заблокирована одним из условий:

- в данный момент обслуживается запрос прерывания равного или более высокого уровня приоритета;
- текущий машинный цикл – не последний в цикле выполняемой команды;
- выполняется команда RETI или любая команда, связанная с обращением к регистрам IE или IP.

Примечание. Если флаг прерывания был установлен, но по одному из перечисленных условий не получил обслуживания и к моменту окончания блокировки уже был сброшен, то запрос прерывания теряется.

По аппаратно сформированному коду команды LCALL система прерывания помещает в стек содержимое программного счётчика PC и загружает в PC адрес вектора прерывания соответствующей подпрограммы обслуживания. По этому адресу должна быть расположена команда безусловного перехода JMP к начальному адресу подпрограммы обслуживания прерывания. Эта подпрограмма в случае необходимости должна начинаться командами записи в стек PUSH слова состояния программы PSW, аккумулятора A, расширителя аккумулятора B, указателя данных DPTR и т.д. и заканчиваться командами восстановления из стека POP. Подпрограммы обслуживания прерывания обязательно завершаются командой RETI, по которой в программный счётчик перезагружается из стека сохранённый адрес возврата в основную программу. Команда RET также возвращает управление, но при этом не снимает блокировку прерывания.

4. ОСОБЕННОСТИ МК C8051F064 SILABS

Микроконтроллер (МК) C8051F064 представляет собой полностью интегрированную на одном кристалле систему для обработки смешанных (аналого-цифровых) сигналов, которая имеет 59 цифровых входа/выхода, а также два встроен-

енных 16-разрядных АЦП с производительностью 1 млн. преобразований в секунду. Отличительные особенности семейства МК C8051 F06х:

- Высокопроизводительное микропроцессорное ядро CIP-51 с конвейерной архитектурой, совместимое со стандартом 8051 (максимальная производительность – 25 MIPS).
- Два встроенных 16-разрядных АЦП (производительность – 1 млн. преобразований в секунду) с контроллером прямого доступа к памяти.
- Встроенные средства отладки, обеспечивающие внутрисистемную, «неразрушающую» отладку в режиме реального времени.
- 64 Кбайта Flash-памяти, программируемой внутрисистемно.
- 4352 (4096 + 256) байт встроенного ОЗУ.
- Интерфейс внешней памяти данных с доступным адресным пространством 64 Кбайта.
- Аппаратно реализованные последовательные интерфейсы I²C/SMBus, SPI и два УАПП.
- Пять 16-разрядных таймеров общего назначения.
- Программируемый массив счетчиков/таймеров (ПМС) с шестью модулями захвата/сравнения.
- Встроенные сторожевой таймер, схема слежения за напряжением питания и датчик температуры.

МК имеет встроенные схему слежения за напряжением питания, сторожевой таймер, тактовый генератор и представляют собой, таким образом, функционально-законченную систему на кристалле. Все аналоговые и цифровые периферийные модули могут включаться/отключаться и настраиваться программой пользователя. Имеется возможность внутрисхемного программирования Flash-памяти, что обеспечивает долговременное (энергонезависимое) хранение данных, а также позволяет осуществлять обновление программного обеспечения в готовых изделиях.

Встроенный интерфейс JTAG позволяет производить «неразрушающую» (не используются внутренние ресурсы) внутрисхемную отладку в режиме реального времени, используя МК, установленные в конечное изделие. Средства отладки обеспечивают проверку и модификацию памяти и регистров, расстановку точек останова и временных меток, пошаговое исполнение программы, а также поддерживают команды запуска и остановки. В процессе отладки с использованием интерфейса JTAG все аналоговые и цифровые периферийные модули полностью сохраняют свою работоспособность.

МК предназначен для работы в промышленном температурном диапазоне (-45°C...+85°C) при напряжении питания 2,7В...3,6В. МК выпускаются в 100-выводных корпусах типа TQFP.

По результатам исследований Annual Creativity in Electronics (ACE) Awards 2005 микроконтроллер C8051F064 признан «Продуктом года» среди микросхем для аналоговой обработки сигналов [9].

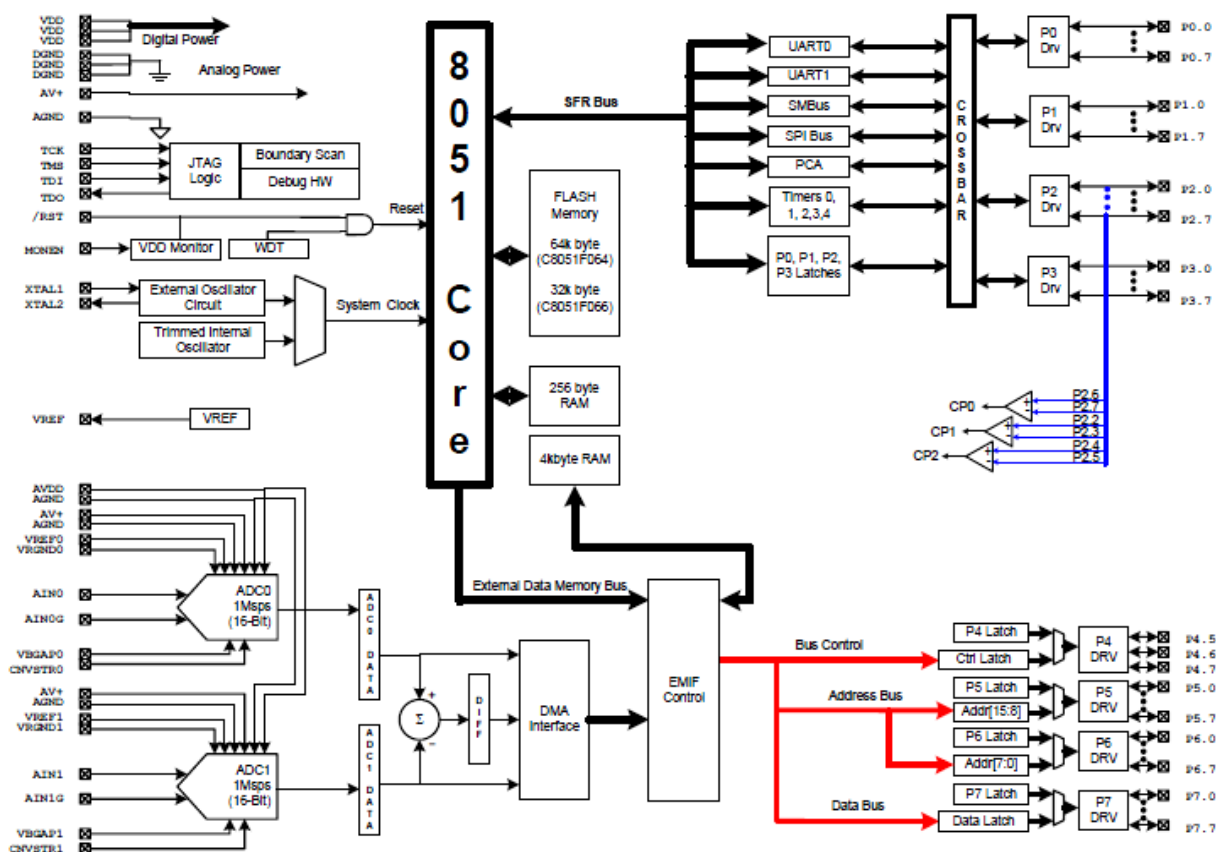


Рис. 3. Структурная схема МК C8051F064

4.1. Процессорное ядро CIP-51

4.1.1. Полностью 8051-совместимая архитектура

МК семейства C8051F06x используют разработанное фирмой Cygnal (приобретённой в 2003 году компанией Silicon Laboratories) процессорное ядро CIP-51, которое по системе команд полностью совместимо с ядром семейства x51. Для разработки программного обеспечения могут использоваться стандартные 803x/805x ассемблеры и компиляторы. Ядро содержит всю периферию, соответствующую стандарту 8052, включая пять 16-разрядных таймеров/счетчиков, два полнодуплексных УАПП, 256 байт внутреннего ОЗУ, 128 байт адресного пространства регистров специального назначения, а также адресуемые побитно порты ввода/вывода.

4.1.2. Улучшенная производительность

CIP-51 использует конвейерную архитектуру, что существенно повышает скорость выполнения команд по сравнению со стандартной архитектурой 8051. В стандартных МК с архитектурой 8051 все команды, кроме MUL и DIV, исполняются за 12 или 24 системных тактовых цикла при максимальной тактовой частоте 12...24 МГц. МК с ядром CIP-51 исполняют 70% своих команд за один или два системных тактовых цикла, и только четыре команды требуют более четырех системных тактовых циклов.

Система команд CIP-51 состоит из 109 команд, которые требуют от одного до восьми системных тактовых циклов:

Количество команд	26	50	5	14	7	3	1	2	1
Количество системных тактовых циклов	1	2	2/3	3	3/4	4	4/5	5	8

При работе на тактовой частоте 25 МГц производительность ядра CIP-51 может достигать 25 MIPS. На рис. 4 показана пиковая производительность различных 8-разрядных МК, работающих на максимально возможных для них частотах.

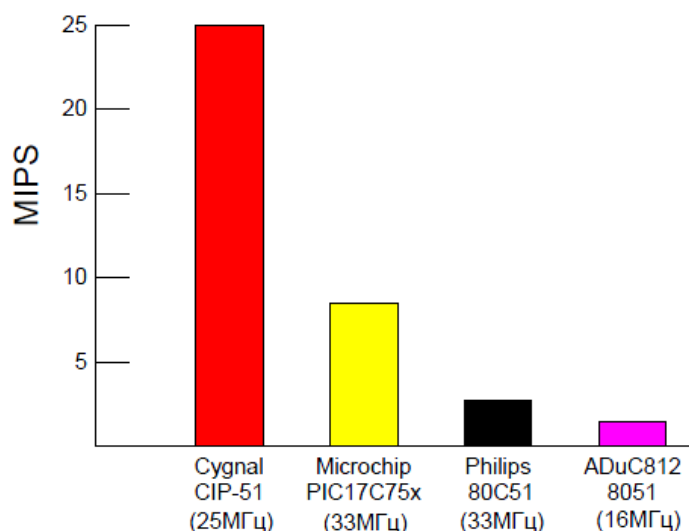


Рис. 4. Максимальная производительность различных микроконтроллеров

4.1.3. Дополнительные функции

МК C8051F064 имеет ряд следующих важных особенностей, которые позволяют улучшить общую производительность и упростить использование МК в конечных приложениях.

22 источника прерываний позволяют многочисленным аналоговым и цифровым периферийным модулям прерывать работу МК. Система управления прерываниями требует меньшего вмешательства со стороны программы, что улучшает ее производительность. Дополнительные источники прерываний очень полезны при построении многозадачных систем, работающих в режиме реального времени.

Имеется семь источников сброса: встроенная схема слежения за напряжением питания, сторожевой таймер, детектор исчезновения тактирования, компаратор нуля, принудительный программный сброс, входной сигнал CNVSTR2 и вывод /RST. Этот вывод является двунаправленным, т.е. может быть как входом внешнего сигнала сброса, так и выходом сигнала сброса, сгенерированного внутри МК схемой слежения за питанием. Любой источник сброса, за исключением схемы слежения за питанием и входного вывода сброса, может быть отключен программно (для включения/отключения схемы слежения за питанием используется вывод

MONEN). Сторожевой таймер может быть включен после сброса типа POR (сброс при включении питания) в процессе инициализации МК.

МК имеет внутренний автономный тактовый генератор, который после сброса используется как источник тактовых импульсов по умолчанию. При необходимости можно “на лету” подключить внешний тактовый генератор, который для генерации тактовых импульсов использует кварцевый или керамический резонатор, конденсатор, RC-цепочку или внешний источник импульсов. В приложениях с пониженным энергопотреблением крайне полезным может быть режим работы МК с медленным (мало потребляющим) внешним кварцевым генератором с периодическим переключением на быстрый (до 25 МГц) внутренний генератор.

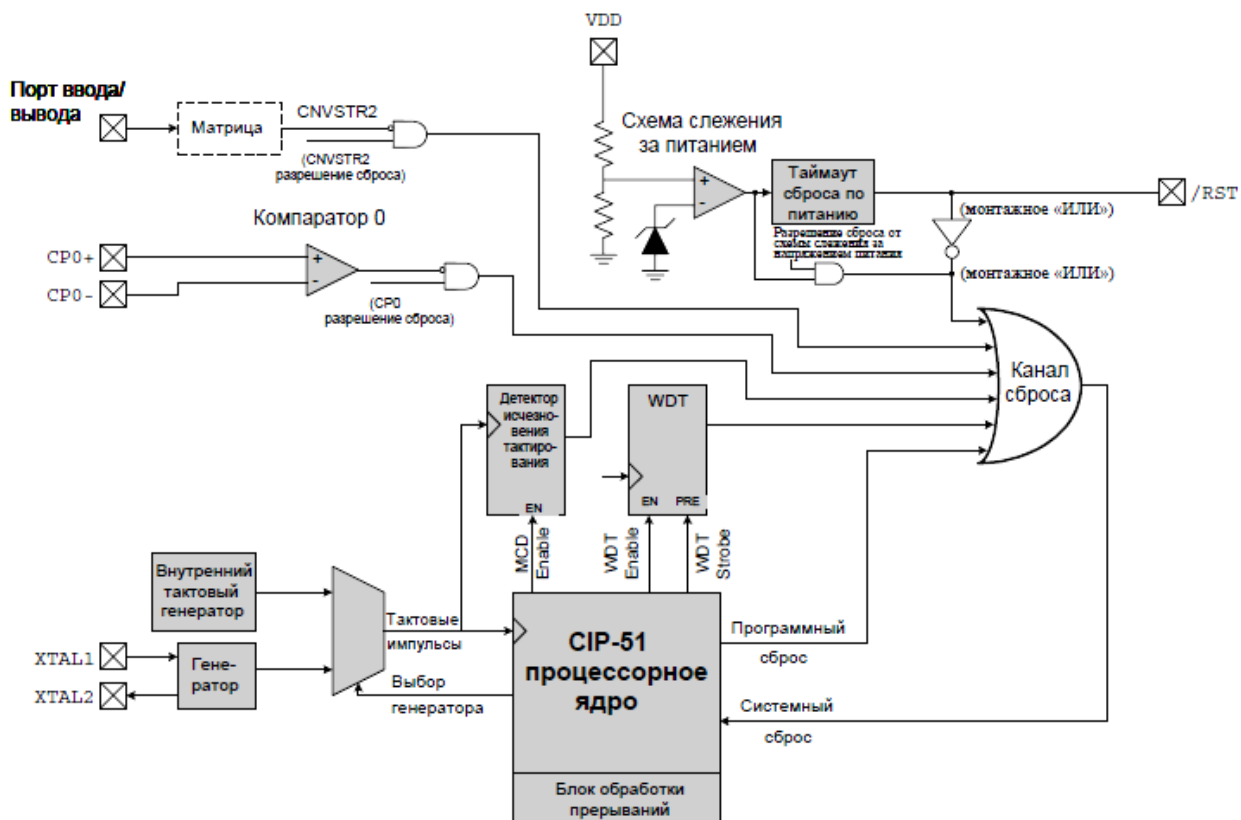


Рис. 5. Структурная схема модуля тактирования и сброса

4.1.4. Встроенная память

CIP-51 имеет стандартную (8051) структуру адресного пространства памяти программ и данных. В состав памяти входит ОЗУ объемом 256 байт. Младшие 128 байт доступны в режиме как прямой, так и косвенной адресации. Из них первые 32 байта адресуются как четыре банка регистров общего назначения, а следующие 16 байт адресуются побайтно или побитно. Старшие 128 байт ОЗУ имеют двойную конфигурацию. В режиме косвенной адресации осуществляется доступ к старшим 128 байтам ОЗУ общего назначения, а в режиме прямой адресации осуществляется доступ к 128 байтам адресного пространства регистров специального назначения

(SFR). Адресное пространство SFR содержит до 256 страниц SFR. Таким образом, МК на основе CIP-51 могут использовать множество регистров SFR для управления и настройки различных периферийных модулей, имеющихся в данных МК.

МК C8051F064 дополнительно имеет встроенный блок 4-Кбайтного ОЗУ. К этому встроенному 4-Кбайтному блоку памяти можно обращаться во всем диапазоне адресов 64 Кбайтной внешней памяти данных (с перекрытием адресов по 4Кбайтным границам). МК C8051F064 имеет также интерфейс внешней памяти (external memory interface – EMIF) для доступа к внешней памяти данных или к периферийным модулям, отображаемым на эту память. На адресное пространство внешней памяти данных может быть отображена либо только встроенная память, либо только внешняя память, либо их комбинация (адреса до 4Кбайт относятся к встроенной памяти, адреса свыше 4Кбайт относятся к EMIF). EMIF может работать с мультиплексированными и не мультиплексированными шинами адреса/данных.

Память программ МК состоит из 64 Кбайт Flash-памяти. Эта память может перепрограммироваться секторами по 512 байт, не требуя при этом специального внешнего напряжения программирования. В МК C8051F064 1024 байт с адресами от 0xEC00 до 0xFFFF зарезервированы для нужд производителя. Во всех МК имеется также дополнительный 128-байтный сектор с адресами от 0x10000 до 0x1007F, который может использоваться в качестве небольшой таблицы программных констант. На рис. 6 приведена карта распределения памяти МК.

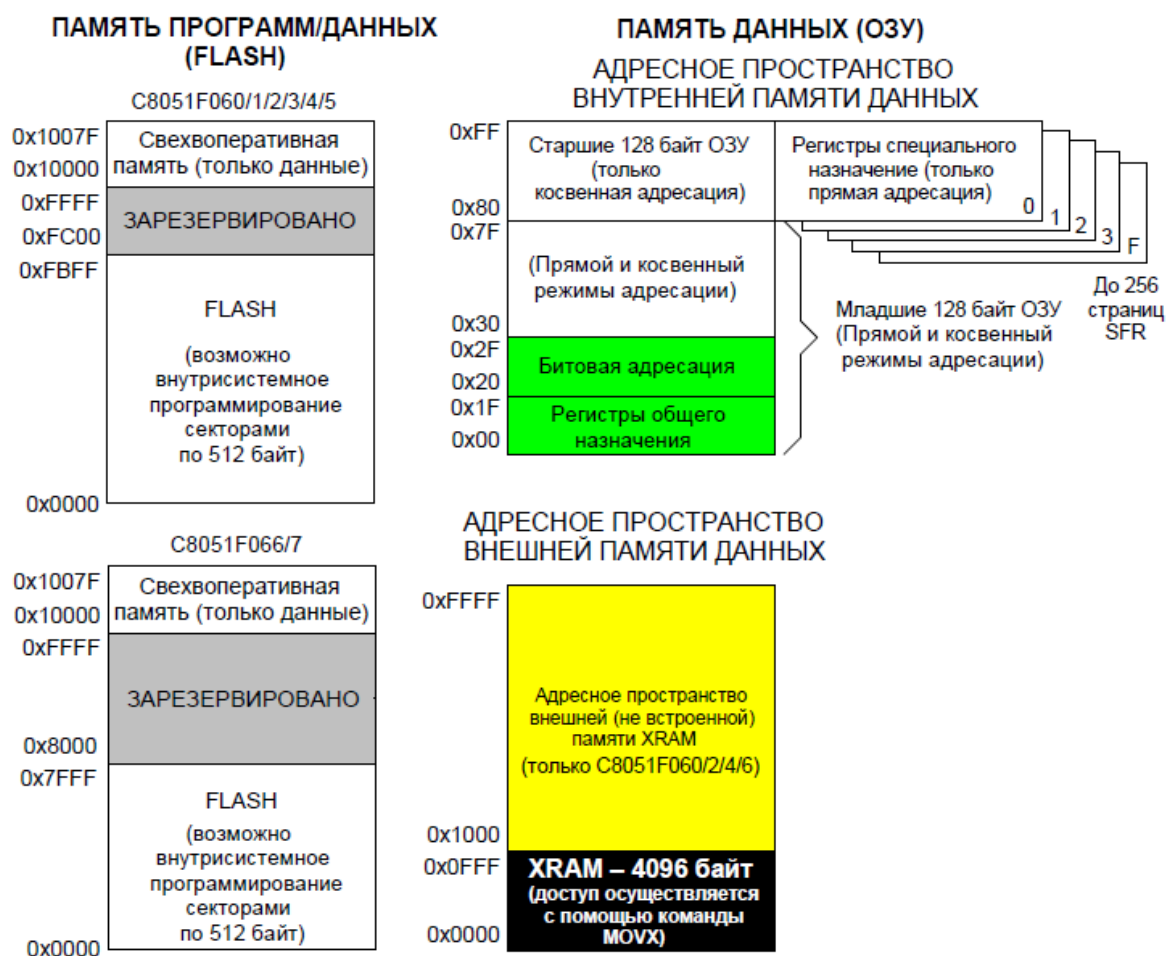


Рис. 6. Карта распределения памяти

4.1.5. JTAG отладчик и интерфейс граничного сканирования

МК C8051F064 имеет встроенный интерфейс граничного сканирования и отладчик, которые посредством 4-х проводного интерфейса JTAG позволяют осуществлять в режиме реального времени «неразрушающую» (не используются внутренние ресурсы) внутрисхемную отладку, используя МК, установленный в конечное изделие. Посредством JTAG интерфейса, полностью совместимого с протоколом IEEE 1149.1, осуществляется граничное сканирование, которое используется для тестирования и производственных испытаний.

Средства отладки фирмы Silicon Labs поддерживают проверку и модификацию памяти и регистров, расстановку точек останова и временных меток, контроль стека, пошаговую отладку. При этом не требуется никаких специальных дополнительных ОЗУ, памяти программ, таймеров или каналов связи. Во время отладки все цифровые и аналоговые периферийные модули не отключаются и работают корректно. При остановке МК в точке останова или при пошаговой отладке работа всех периферийных модулей (кроме АЦП и SMBus) блокируется, что необходимо для удержания их в режиме синхронизации с выполнением команд.

По сравнению со стандартными симуляторами такой способ разработки и отладки встроенных систем обеспечивает следующие преимущества:

- не требуется отладочный кристалл;
- не используются специализированные кабели;
- не требуется использовать разъем для установки МК на плату.

Отладочная среда фирмы Silicon Labs обеспечивает удобство работы с прецизионными аналоговыми периферийными модулями и при этом не ухудшает их производительности.

4.1.6. Программируемые порты ввода/вывода и матрица соединений

МК имеет три стандартных для архитектуры 8051 порта (0, 1 и 2), четыре дополнительных 8-разрядных порта (3, 5, 6 и 7) и один 3-разрядный порт (4), т.е. всего 59 линий ввода/вывода общего назначения. Порты функционируют в соответствии со стандартом 8051 с некоторыми дополнительными возможностями.

Каждый вывод порта может быть настроен либо как цифровой вход-выход, либо как выход с открытым стоком. Кроме этого, возможно общее отключение подтягивающих резисторов (которые в стандартной архитектуре 8051 обычно нельзя отключать), что позволяет еще более снизить энергопотребление в критичных к этому параметру приложениях.

Наиболее важным усовершенствованием является цифровая коммутационная матрица, которая позволяет необходимым образом соединять внутренние цифровые системные ресурсы с выводами портов ввода/вывода P0, P1, P2 и P3 (см. рис.7). При этом, в отличие от МК со стандартными мультиплексированными цифровыми портами ввода/вывода, возможны любые комбинации для МК в любом корпусе.

При помощи регистров управления матрицей на выводы портов могут быть выведены сигналы от внутренних таймеров/счетчиков, от последовательных интерфейсов, аппаратные прерывания, входной сигнал запуска АЦП, выходы компараторов и др. Это позволяет пользователю выбрать точную комбинацию связей между портами ввода/вывода общего назначения и цифровыми ресурсами, необходимым для каждого конкретного приложения.

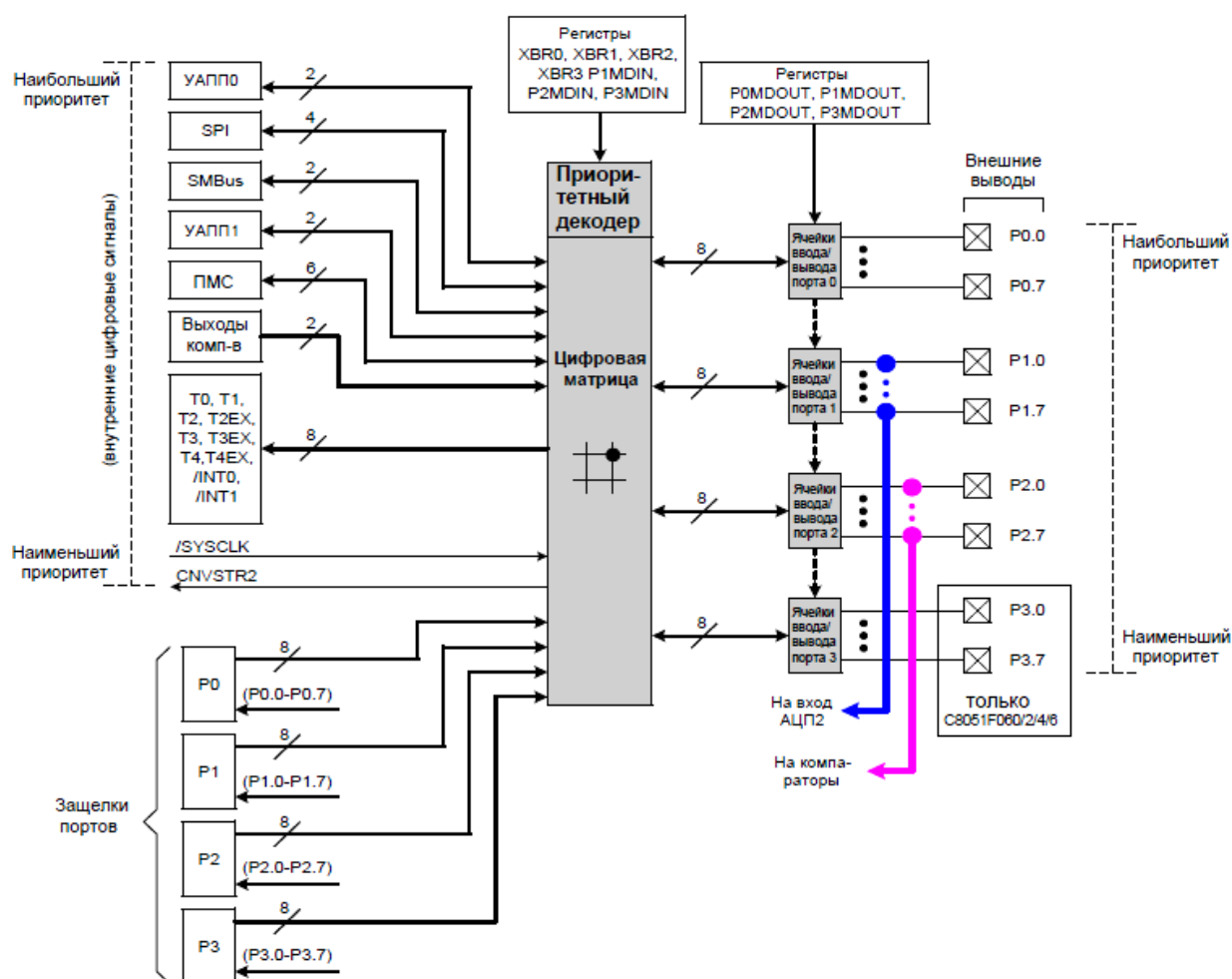


Рис. 7. Структурная схема цифровой матрицы

4.1.7. Программируемый массив счетчиков (ПМС)

МК C8051F064 кроме пяти 16-разрядных таймеров/счетчиков общего назначения имеет внутренний программируемый массив счетчиков (ПМС) (рис. 8). ПМС состоит из специального 16-разрядного таймера/счетчика временных интервалов с шестью программируемыми модулями захват/сравнение. В качестве тактового сигнала для этого счетчика могут использоваться:

- сигнал системного тактового генератора с частотой, поделённой на 12;
- сигнал системного тактового генератора с частотой, поделённой на 4;
- сигнал переполнения таймера 0;
- сигнал от внешнего входа тактирования (ECI – external clock input);
- системный тактовый сигнал;
- сигнал внешнего генератора с частотой, делённой на 8.

Каждый модуль захват/сравнение может быть настроен на работу в одном из шести режимах:

- захват, управляемый фронтом сигнала;
- программный таймер;
- высокоскоростной выход;
- выход заданной частоты;
- 8-разрядный широтно-импульсный модулятор;
- 16-разрядный широтно-импульсный модулятор.

Входы/выходы модулей захват/сравнение ПМС и внешний вход тактирования (ECI) соединены с портами ввода/вывода МК через цифровую коммутирующую матрицу.

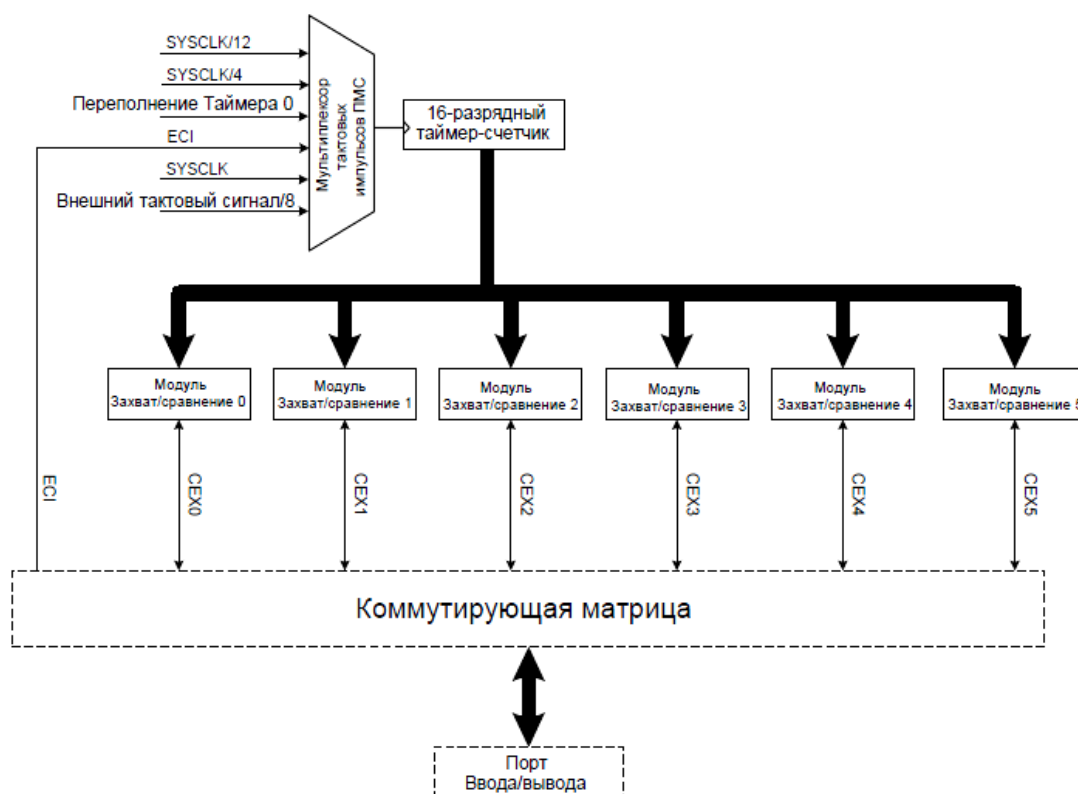


Рис. 8. Структурная схема модуля ПМС

4.1.8. Последовательные порты

В МК C8051F064 встроены следующие последовательные интерфейсы:

- два усовершенствованных полнодуплексных УАПП;
- усовершенствованный SPI;
- I2C/SMBus.

Каждый из этих интерфейсов реализован на аппаратном уровне и широко использует прерывания, требуя лишь незначительного вмешательства со стороны программы пользователя. Эти интерфейсы не имеют общих ресурсов, таких как

таймеры, прерывания или порты ввода/вывода, поэтому все они могут использоваться одновременно.

4.1.9. 16-разрядный аналого-цифровой преобразователь

МК C8051F064 имеет два встроенных 16-разрядных АЦП последовательного приближения (АЦП0 и АЦП1), которые могут использоваться по отдельности в однофазном режиме или совместно в дифференциальном режиме. С помощью интерфейса прямого доступа к памяти (DMA), АЦП0 и АЦП1 могут непосредственно обращаться к внутреннему или внешнему ОЗУ. При максимальной производительности 1 млн. преобразований в секунду эти АЦП обеспечивают 16-битную точность преобразования с двумя уровнями нелинейности. АЦП0 и АЦП1 могут использовать либо специальный внутренний источник опорного напряжения (ИОН), либо внешний ИОН.

Управление АЦП осуществляется при помощи регистров специального назначения. Имеется возможность отключения этих АЦП с целью уменьшения энергопотребления.

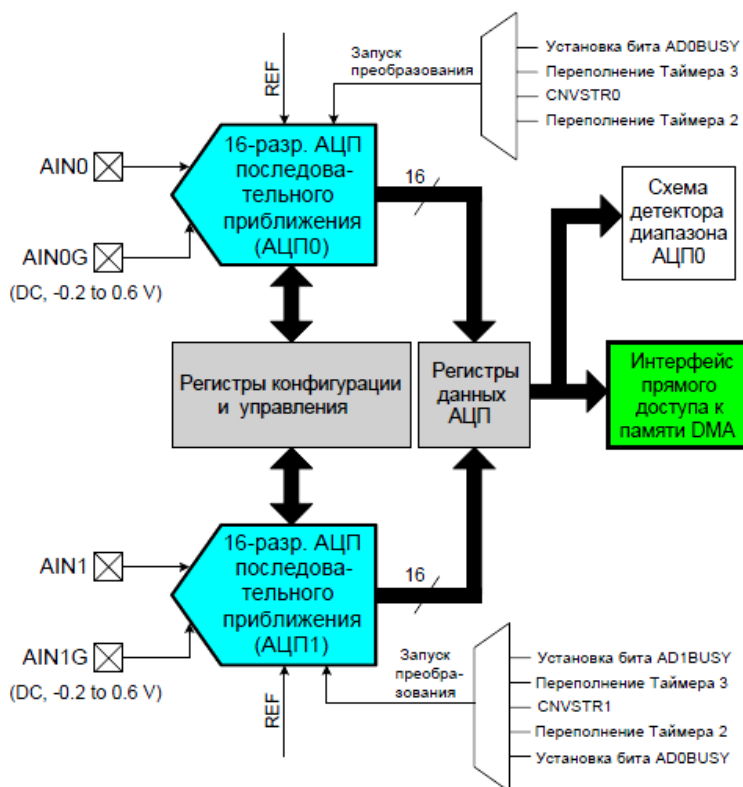


Рис. 9. Структурная схема 16-разрядного АЦП

Преобразование может быть запущено четырьмя способами: командой в программе, при переполнении таймера 2, при переполнении таймера 3 или внешним входным сигналом. Такая гибкость позволяет запускать преобразование при возникновении определенных программных событий, по сигналам от внешних устройств или периодически при переполнении таймера. Оба АЦП могут функцио-

низовать как независимо друг от друга, так и синхронно, что позволяет осуществлять преобразования одновременно. По окончании преобразования устанавливается специальный бит состояния и инициируется прерывание, если оно разрешено, после чего полученное 16-разрядное слово данных записывается в два регистра специального назначения. Также имеется интерфейс DMA, который позволяет получать результаты преобразований АЦП и сохранять их непосредственно во внутреннем или внешнем ОЗУ.

АЦПО содержит также детектор диапазона, который можно настроить таким образом, чтобы генерировать прерывание лишь при попадании или непопадании результата преобразования в заданный диапазон значений (окно). АЦПО может непрерывно отслеживать сигнал в фоновом режиме, но не прерывать МК до тех пор, пока преобразованные данные не окажутся в пределах заданного диапазона.

4.1.10. Аналоговые компараторы

МК C8051F064 имеет три встроенных аналоговых компаратора. Компараторы допускают программирование гистерезиса и времени отклика. Каждый компаратор может генерировать прерывание по переднему или заднему фронту петли гистерезиса, либо по обоим фронтам. Эти прерывания могут вывести МК из режима остановки, а прерывание от компаратора 0 может также использоваться в качестве источника сброса. Состояние выходов компараторов можно опрашивать программно. Кроме этого, с помощью коммутирующей матрицы можно вывести сигналы с выходов компараторов на внешние порты ввода/вывода. Когда компараторы не используются, их можно перевести в режим пониженного энергопотребления.

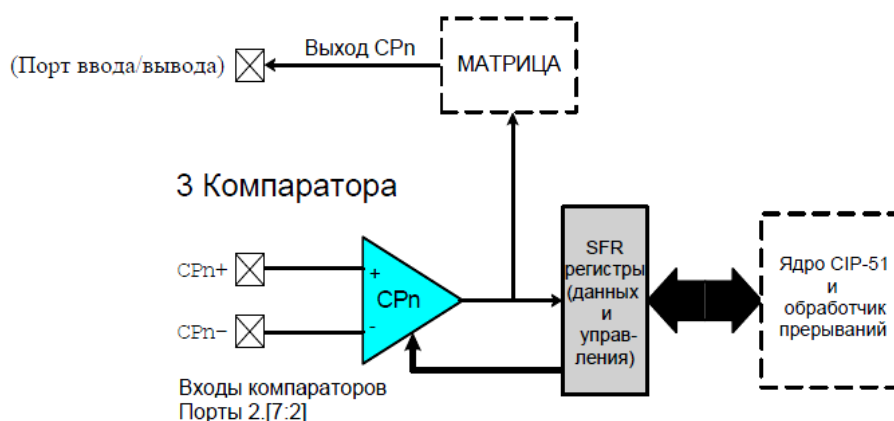


Рис. 10. Структурная схема компаратора

5. ЛАБОРАТОРНАЯ УСТАНОВКА C8051F064EK EVALUATION KIT

В состав системы начального освоения C8051F064EK EVALUATION KIT входят:

- плата начального освоения C8051F064EK;
- кабель USB;

интегрированная среда разработки IDE Silabs (ассемблер, линкер, С-компилятор);

документация;

файлы определения регистров и исходные тексты примеров;

демонстрационная программа работы с АЦП;

инструкция по эксплуатации.

5.1. Плата начального освоения C8051F064EK

На рис. 11 приведена блок-схема основных элементов платы начального освоения **C8051F064EK**. На плате находится функционально завершенная система сбора данных, использующая микроконтроллер **C8051F064**. На плате также расположены два USB порта, обеспечивающие подключение персонального компьютера (порт передачи данных (DATA) и отладочный порт (DEBUG)). Для подключения DATA порта используется микроконтроллер–преобразователь UART – USB на основе МК CP2101 (Silicon Labs). DEBUG порт используется в системе отладки IDE Silabs (Self-Demo/ IDE Debug port). Питание платы осуществляется через коннектор USB.

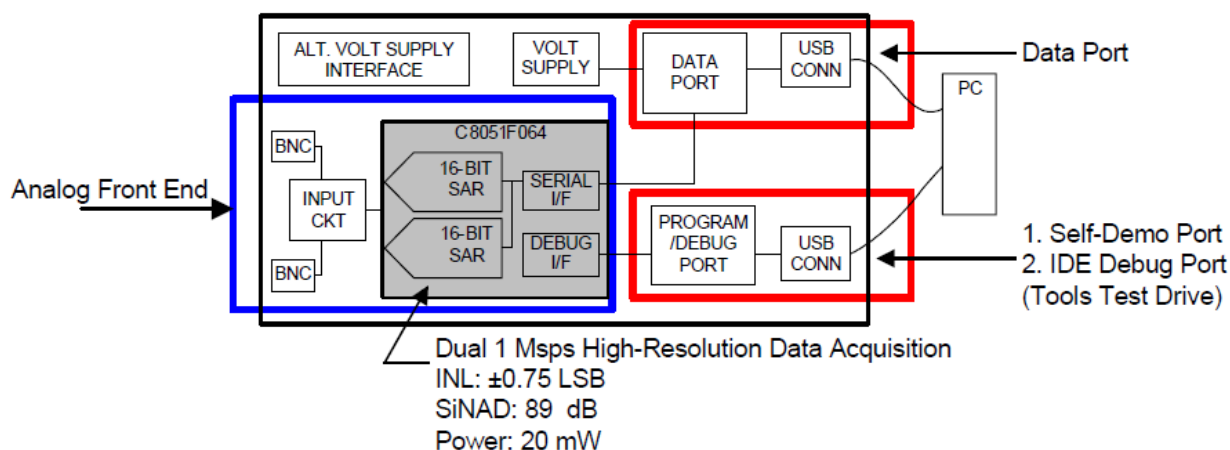
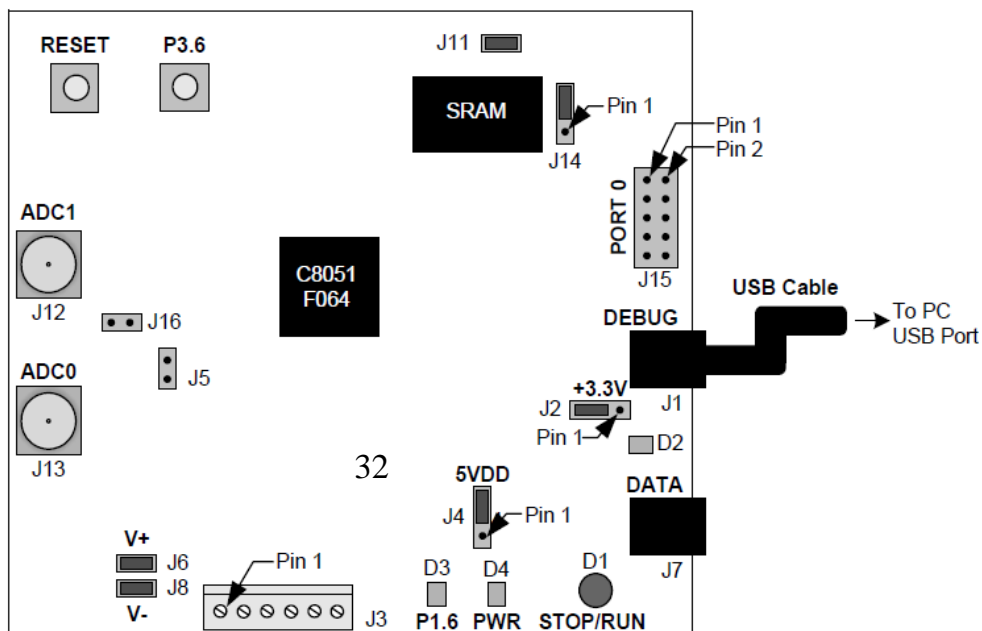


Рис. 11. Диаграмма платы начального освоения **C8051F064EK** [11]

На рис. 12 показаны, помимо прочего, коннекторы J, позволяющие подклю-



чать периферийные устройства и конфигурировать отладочную плату:

Рис. 12. Внешний вид отладочного модуля [11]

J1 – отладочный коннектор USB для подключения платы к PC, программирования МК и проведения отладки IN-SYSTEM;

J2 – выбор источника питания платы;

J3 – питание аналоговый устройств ввода/вывода;

J4 – выбор внешнего опорного источника;

J5 – вход внешнего запуска аналого-цифрового преобразования;

J7 – порт передачи данных на USB;

J6, J8 – выбор источника питания для операционных усилителей АЦП;

J11, J14 – подключение внешнего ОЗУ;

J12, J13 – BNC входы АЦП1 и АЦП0;

J15 – разъем подключения 0 порта P0;

J16 – переключатель АЦП в дифференциальный режим измерения.

5.2. Источник системной опорной частоты

МК C8051A064 изначально использует внутренний источник. После сброса внутренний генератор переключается по умолчанию на частоту 3,0625 МГц, но может программно быть переключен на другие частоты. Для многих приложений требуется более высокая стабильность тактовой частоты, тогда можно использовать внешний резонатор с частотой 22,1184 МГц. Конфигурирование системного генератора даётся в техническом описании МК [10, 11].

5.3. Переключатели и светодиоды

На отладочной плате установлены 2 кнопки. Переключатель SW1 осуществляет функцию сброса и подключен к входу *RESET* МК C8051F064. Нажатие на SW1 приводит к аппаратной перезагрузке МК. Кнопка SW2 подключена к входу порта ввода/вывода (GPIO) P3.7. Нажатие на SW2 приводит к появлению низкого уровня на входе порта.

На плате установлены также 4 светодиода:

D1 – двухцветный светодиод-индикатор режимов Run/Stop показывает соединение отладочной платы с персональным компьютером;

D2 – красный светодиод показывает наличие связи Data-порта с USB;

D3 – зелёный светодиод подключен к входу порта P1.6;

D4 – красный светодиод «PWR» показывает наличие питания.

6. ВВЕДЕНИЕ В IDE SILABS

IDE Silabs фирмы Silicon Laboratories – интегрированная среда разработки программного обеспечения для однокристальных микроконтроллеров семейства Intel 8051 и его клонов. Она включает в себя всё, что нужно для создания, редактирования, компиляции, трансляции, компоновки, загрузки и отладки программ:

- стандартный интерфейс Windows,
- полнофункциональный редактор исходных текстов с выделением синтаксических элементов цветом,
- организатор проекта,
- транслятор с языка C,
- ассемблер,
- отладчик,
- встроенную справочную систему.

Первый этап разработки программы – запись её исходного текста на каком-либо языке программирования.

Затем производится компиляция или трансляция его в коды из системы команд микроконтроллера, используя транслятор или ассемблер. Трансляторы и ассемблеры – прикладные программы, которые интерпретируют текстовый файл, содержащий исходный текст программы, и создают объектные файлы, содержащие объектный код.

После компоновки объектных модулей наступает этап отладки программы, устранения ошибок, оптимизации и тестирования программы.

IDE Silabs объединяет все этапы разработки прикладной программы в единый рекурсивный процесс, когда в любой момент времени возможен быстрый возврат к любому предыдущему этапу.

IDE Silabs имеет следующие компоненты: кросс-компилятор C51, макроассемблер A51, компоновщик L51, отладчик/симулятор.

6.1. Оптимизирующий кросс-компилятор C51 (Keil C51)

Язык C – универсальный язык программирования, который обеспечивает эффективность кода, элементы структурного программирования и имеет богатый набор операторов. Универсальность, отсутствие ограничений реализации делают язык C удобным и эффективным средством программирования для широкого разнообразия задач. Множество прикладных программ может быть написано легче и эффективнее на языке C, чем на других более специализированных языках.

C51 – полная реализация стандарта ANSI (Американского национального института стандартов), насколько это возможно для архитектуры Intel 8051. C51 генерирует код для всего семейства микроконтроллеров Intel 8051. Транслятор сочетает гибкость программирования на языке C с эффективностью кода и быстродействием ассемблера.

Использование языка высокого уровня C имеет следующие преимущества над программированием на ассемблере:

- не требуется глубокого знания системы команд процессора, элементарное знание архитектуры Intel 8051 желательно, но не необходимо;
- распределение регистров и способы адресации полностью управляются транслятором;
- лучшая читаемость программы, используются ключевые слова и функции, которые более свойственны человеческому мышлению;
- время разработки программ и их отладки значительно короче в сравнении с программированием на ассемблере;
- библиотечные файлы содержат много стандартных подпрограмм, которые могут быть включены в прикладную программу;
- используя модульные методы программирования, можно многократно использовать уже имеющиеся программы в новых проектах.

Версия **Keil C51**, поставляемая с отладочным модулем, автоматически устанавливается при инсталляции среды **IDE**. Эта версия накладывает ограничение на компилируемые файлы, позволяя создавать модули размером до 2 КБ. В файле `c51.pdf` в каталоге `\Silabs\MCU\hlp` содержится справочная информация по использованию компилятора, которая может быть вызвана через меню **Help>Keil Compiler Manual**.

6.2. Макроассемблер A51 (Keil Software A51)

Ассемблер A51 совместим с ассемблером ASM51 для всего семейства микроконтроллеров 8051. Ассемблер транслирует символическую мнемонику в перемещаемый объектный код, имеющий высокое быстродействие и малый размер. При отладке или при включенной опции “Include debugging information” этот объектный файл будет содержать полную символическую информацию для отладчика/имитатора или внутрисхемного эмулятора. Макросредства ускоряют разработку и экономят время, поскольку общие последовательности могут быть разработаны только один раз. Ассемблер поддерживает символический доступ ко всем элементам микроконтроллера и перестраивает конфигурацию для каждой разновидности Intel 8051. Англоязычное описание ассемблера содержится в файле `a51.pdf` в каталоге `\Silabs\MCU\hlp`.

6.3. Компоновщик L51 (BL51)

Компоновщик объединяет один или несколько объектных модулей в одну исполняемую программу. Компоновщик размещает внешние и общие ссылки, назначает абсолютные адреса перемещаемым сегментам программ. Он может обрабатывать объектные модули, созданные транслятором C51, ассемблером A51, транслятором PL/M-51 и ассемблером ASM51.

Компоновщик автоматически выбирает соответствующие библиотеки поддержки и связывает только требуемые модули из библиотек. Установки по умолчанию для L51 выбраны так, чтобы они подходили для большинства прикладных программ, но можно определить и заказные установки.

6.4. Отладчик/симулятор

Отладчик исходных текстов используется с транслятором C51, ассемблером A51, транслятором PL/M-51 и ассемблером ASM51. Отладчик/симулятор позволяет

моделировать большинство особенностей Intel 8051 без наличия аппаратных средств. Можно использовать его для проверки и отладки прикладной программы прежде, чем будут изготовлены аппаратные средства. При этом моделируется широкое разнообразие периферийных устройств, включая последовательный порт, внешний ввод-вывод и таймеры.

7. ЗНАКОМСТВО С РАБОТОЙ 8051 F064EK

Выполним этот этап, взяв в качестве примера программу “Blink” из папки \Silabs\MCU\Examples\C8051F06x, которая обеспечивает мигание светодиода на отладочной плате. В файле blink.asm содержится исходный текст программы:

```

/*****
/* Ваша первая ассемблерная C8051F064 программа */
/*****
;-----
; Copyright (C) 2004 Silicon Laboratories, Inc.
; Все права защищены.
; Имя файла: BLINK.ASM
; Применяемые МК: C8051F060/1/2/3/4
; Назначение : Эта программа демонстрирует как отключить watchdog таймер,
; сконфигурировать порт и вывести информацию на вывод порта ввода/вывода.
; NOTES:
;-----
#include (c8051f060.inc) ; Подключает файл определения регистров.
;-----
; EQUATES
;-----
GREEN_LED equ P1.6 ; Определение контакта порта ввода/вывода, к кото-
; рому подключен зеленый светодиод.
;-----
; RESET and INTERRUPT VECTORS
;-----
; Старт программы при сбросе.
; cseg AT 0
; ljmp Main ; Определение стартового перехода при сбросе.
;-----
; Сегмент кода
;-----
Blink segment CODE
rseg Blink ;Switch to this code segment.
using 0 ; Определение банка регистров для использования
; Код программы.
Main: ; Отключение WDT. (Прерывания в этой точке недоступны.)
; Если прерывания доступны, необходимо явно выключить их так чтобы
; второе срабатывание WDTCN occurs no more than four clock
; cycles after the first move to WDTCN.
mov WDTCN, #0DEh
mov WDTCN, #0ADh
; Use SFRs on the Configuration Page
mov SFRPAGE, #CONFIG_PAGE
; Enable the Port I/O Crossbar
mov XBR2, #40h
; Установка P1.6 (LED) в режим цифрового ввода/вывода в push-pull моду.
orl P1MDOUT, #40h
; Выключение светодиода
clr GREEN_LED

```

```

; Программно формируемый цикл задержки.
Loop2:      mov    R7, #03h
Loop1:      mov    R6, #00h
Loop0:      mov    R5, #00h
            djnz   R5, $
            djnz   R6, Loop0
            djnz   R7, Loop1
            cpl    GREEN_LED ;Переключение светодиода инверсией бита(LED).
            jmp    Loop2
;-----
; End of file.
END

```

Прежде чем начать отладку проекта, скопируйте папку \Silabs\MCU\Examples\C8051F06x в свою личную папку.

7.1. Запуск IDE Silabs и открытие файла проекта

IDE Silabs запускается из стартового меню Windows подобно остальным приложениям (рис. 13).

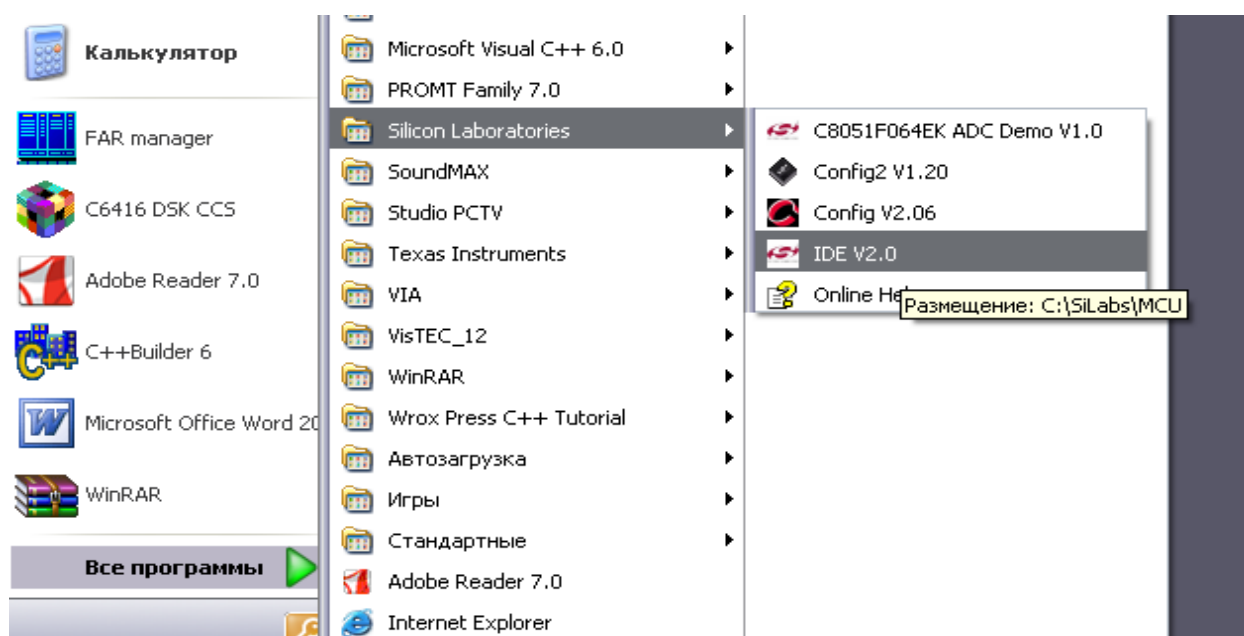


Рис. 13. Запуск программы

Для того, чтобы открыть файл проекта:

- выберите меню **Open Project > Project**.
- В окне диалога **Open File** найдите свой каталог и выберите в нем файл `blink.wsp`. Файл с расширением `.wsp` содержит конфигурацию проекта. Окно отладчика IDE представлено на рис. 14.

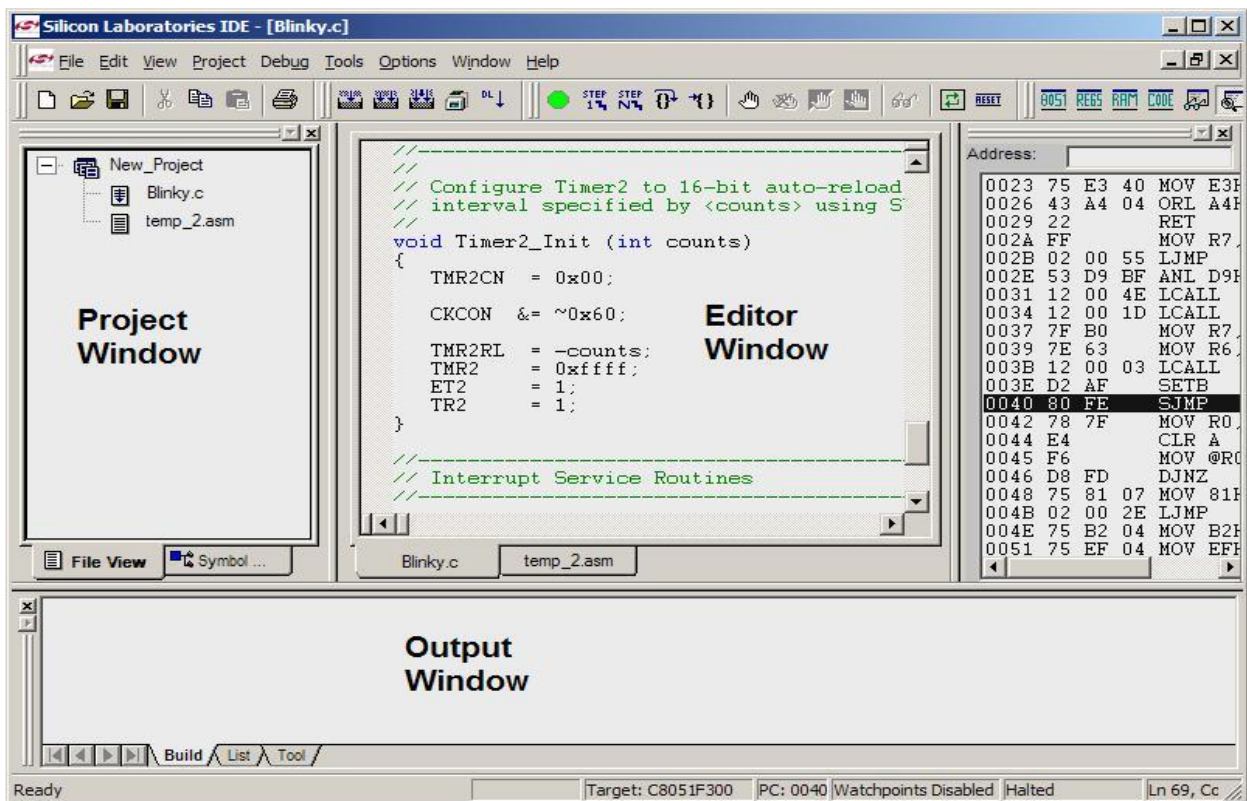


Рис. 14. Окно отладчика IDE Silabs

- В левом окне IDE Silabs находится древовидная структура **Project Window**, отображающая структуру проекта (рис.15).

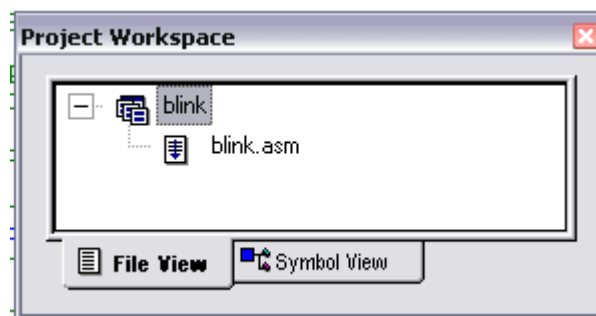


Рис. 15. Диалоговое окно Project Window

- Щелкнув по иконке любого файла из состава проекта, можно получить его листинг в стандартном окне редактирования (рис.16). IDE Silabs загружает и показывает содержание `blink.asm` в окне, где можно редактировать файл. Окно редактирования – полнофункциональный редактор исходного текста, предлагающий такие возможности, как высвечивание синтаксических элементов.

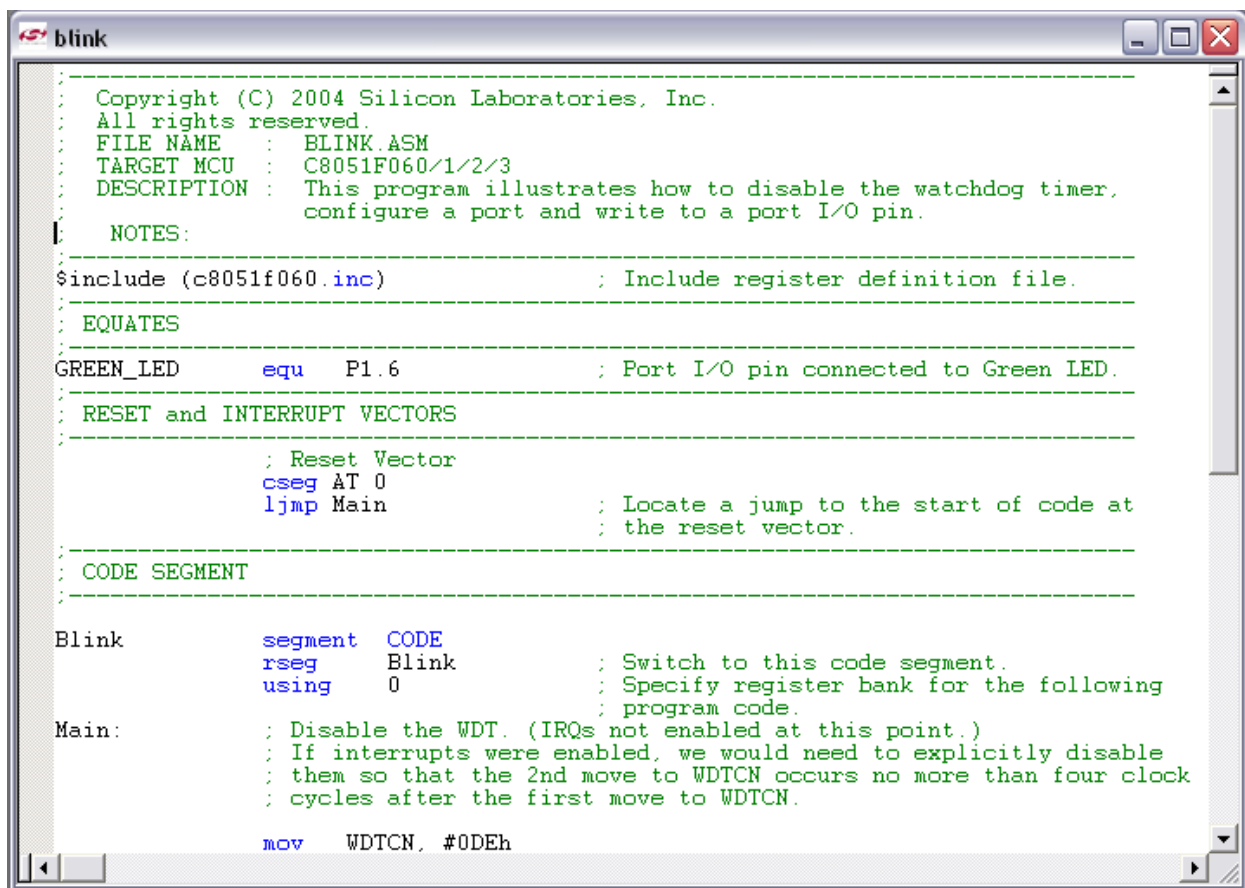


Рис. 16. Окно редактирования

- Выбрав пункт **Build/Make Project** или **Rebuild Project** из меню **Project**, произведите компиляцию и сборку проекта. Результат будет отображаться в окне сообщений с предупреждениями (WARNINGS) и ошибками (ERRORS) (рис. 17).

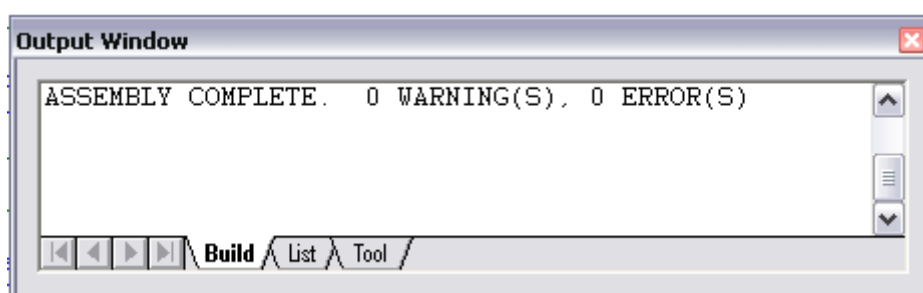


Рис.17. Окно сообщений

- При необходимости настройте опции связи через меню **Options>Connection Options**. В появившемся окне «**Connection Options**» (рис. 18) установите для данной отладочной платы параметры **USB serial adapter** и **JTAG** отладчик.

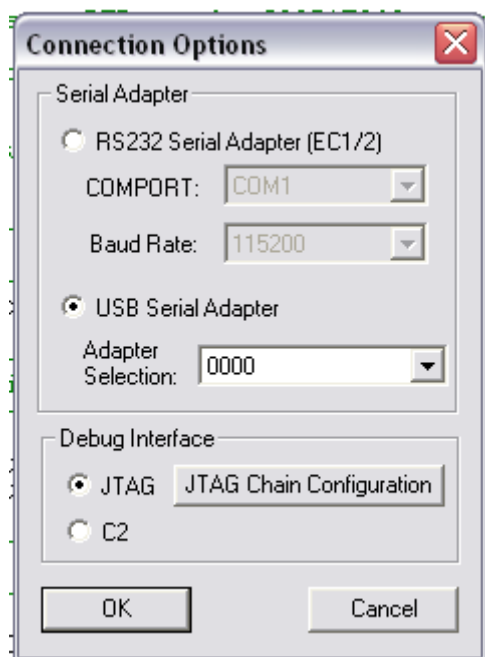


Рис. 18. Окно настройки параметров связи

- Выбрав пункт меню **Debug>Connect**, проведите связь с отладочным модулем.
- Выбрав пункт меню **Debug>Download Object File**, загрузите объектный файл проекта в область Flash памяти программ МК.

Далее, воспользовавшись отладочными возможностями, произведите сеанс пошаговой отладки проекта внутри системы (In-system). На рис. 19 показано, как выглядит экран отладчика IDE Silabs при выполнении программы. Вы можете использовать отладчик, чтобы перемещаться по программе. Выберите **Reset** из меню **Debug** (эта команда сбросит моделируемый процессор) и выберите **Step Into** и **Step Over** из меню **Debug**. Выберите **Start** из меню **Debug**. Выберите **Run** из меню **Debug** или нажмите кнопку. Команды **Step** позволяют “шагать” по каждой строке исходного текста. Текущая команда высвечивается на каждом шаге. **Step Into** позволяет войти в вызываемую функцию, **Step Over** – перешагнуть через неё, не входя во внутрь. Прodelайте эти операции. Для завершения работы с отладчиком в любой момент времени Вы можете выбрать **Disconnect** из меню **Debug** и возвратиться в режим редактирования

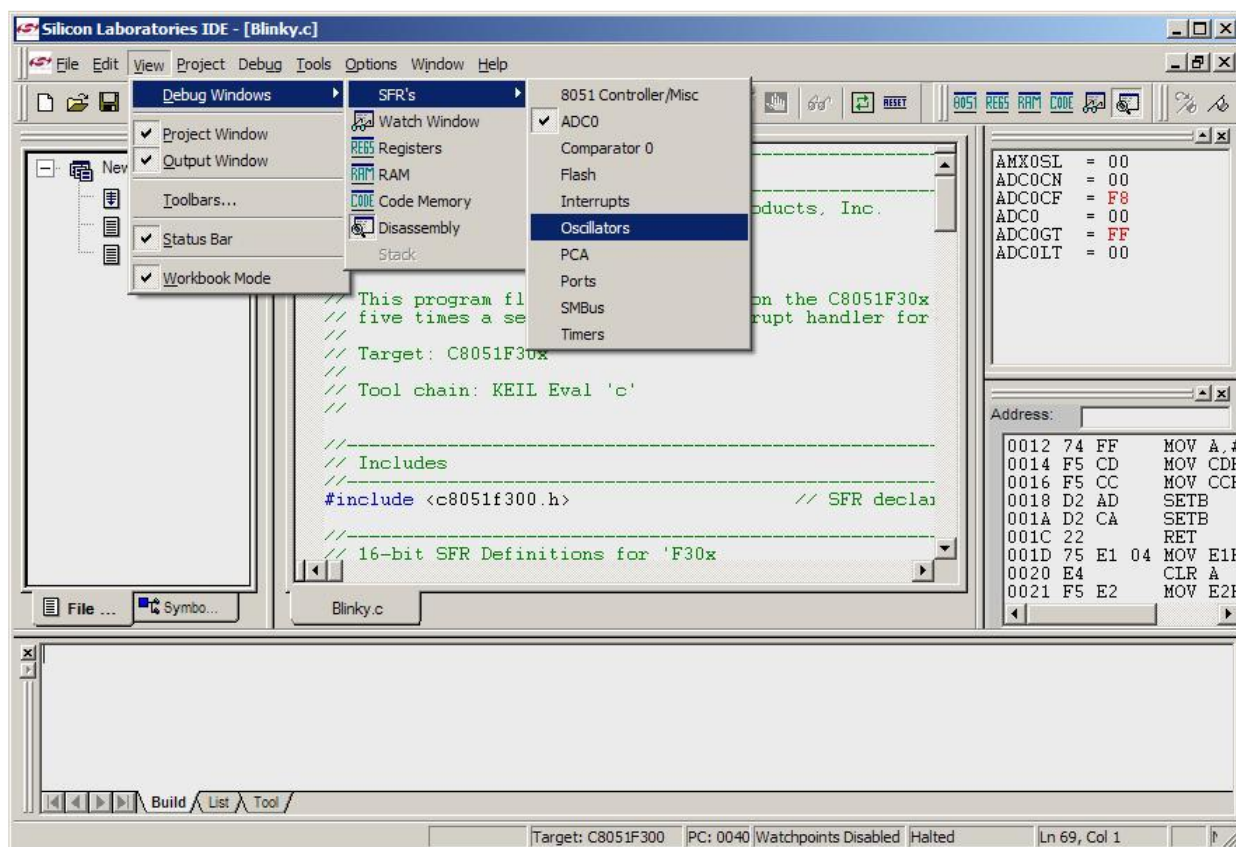


Рис. 49. Экран отладчика при выполнении программы

Предложенная в качестве примера программа организует мигание светодиода, используя метод программного формирования задержек.

7.2. Запуск IDE Silabs и создание файла проекта с использованием компилятора языка Си

Любая работа в IDE Silabs начинается с создания нового файла проекта. Файл проекта содержит имена всех исходных файлов, связанных с проектом, а также установки компиляции, трансляции и связывания файлов, чтобы генерировать выполняемую программу. Программа мигания светодиода на языке Си выглядит следующим образом:

```
//-----
// Blinky.c
//-----
// Copyright (C) 2004 Silicon Laboratories, Inc.
//
// AUTH: BD
// DATE: 26 SEP 2002
//
```

```

// Эта программа зажигает светодиод на плате C8051F064 через каждые 5 секунд, используя
// прерывания таймера №3. Светодиод подключен к выводу P1.6 порта ввода/вывода C8051F064.
//-----
// Includes
//-----
#include <c8051f060.h>          // Описания SFR регистров
//-----
// 16-bit SFR регистры для 'F06x
//-----
sfr16 RCAP3  = 0xCA;          // Перезагружаемая 16-битовая константа таймера №3
sfr16 TMR3   = 0xCC;          // Счетчик таймера №3
//-----
// Global CONSTANTS
//-----
#define SYSCLK 3062500         // перевод частоты SYSCLK в герцы
sbit LED = P1^6;              // LED: '1' = ON; '0' = OFF
//-----
// Function PROTOTYPES
//-----
void PORT_Init (void);
void Timer3_Init (int counts);
void Timer3_ISR (void);
//-----
// MAIN Routine
//-----
void main (void) {
    // отключение watchdog таймера
    WDTCN = 0xde;
    WDTCN = 0xad;
    PORT_Init ();
    Timer3_Init (SYSCLK / 12 / 10); // Init Timer3 to generate interrupts at a 10 Hz rate.
    EA = 1;                        // enable global interrupts
    SFRPAGE = LEGACY_PAGE;         // Page to sit in for now
    while (1) {                    // spin forever
        }
    }
//-----
// PORT_Init
//-----
//

```

```

// Конфигурирование матрицы соединений портов ввода/вывода
//
void PORT_Init (void)
{
char old_SFRPAGE;
old_SFRPAGE = SFRPAGE;    // Save old SFRPAGE
SFRPAGE = CONFIG_PAGE;    // Switch to configuration page
XBR2  = 0x40;              // Enable crossbar and weak pull-ups
P1MDOUT |= 0x40;          // enable P1.6 (LED) as push-pull output
SFRPAGE = old_SFRPAGE;    // restore SFRPAGE
}
//-----
// Timer3_Init
//-----
//
// Configure Timer3 to auto-reload and generate an interrupt at interval
// specified by <counts> using SYSCLK/12 as its time base.
//
void Timer3_Init (int counts)
{
char old_SFRPAGE;
old_SFRPAGE = SFRPAGE;    // Save old SFRPAGE
SFRPAGE = TMR3_PAGE;      // Switch to Timer 3 page
TMR3CN = 0x00;            // Stop Timer3; Clear TF3; use SYSCLK/12 as timebase
RCAP3  = -counts;         // Init reload values
TMR3   = 0xffff;          // set to reload immediately
EIE2   |= 0x01;           // enable Timer3 interrupts
TR3 = 1;                  // start Timer3
SFRPAGE = old_SFRPAGE;    // restore SFRPAGE
}
//-----
// Interrupt Service Routines
//-----
//-----
// Timer3_ISR
//-----
// This routine changes the state of the LED whenever Timer3 overflows.
//
// NOTE: The SFRPAGE register will automatically be switched to the Timer 3 Page. When an interrupt
// occurs. SFRPAGE will return to its previous setting on exit from this routine.

```

```
//
void Timer3_ISR (void) interrupt 14
{
    TF3 = 0;                // clear TF3
    LED = ~LED;             // change state of LED
}
```

Для того, чтобы создать новый файл проекта:

- выберите меню **Project > New Project**. Когда менеджер проекта открывает файл проекта, окно проекта показывает включенные исходные файлы. В данном случае пока нет никаких исходных файлов. Теперь можно добавить файл `blinky.c` к проекту.
- Выберите меню **Project > Add file**. Откроется диалоговое окно **Add File**. Выберите файл `blinky.c` из списка. Данный проект имеет только один исходный файл. В дальнейшем Ваши проекты, возможно, будут состоять из множества исходных файлов. Диалог **Add File** позволит Вам выбрать и добавить несколько файлов сразу. Для этого используют комбинацию клавиши **Ctrl** и указателя мыши. Когда Вы нажмёте [Open], исходные файлы будут добавлены к проекту в выбранном порядке.

Теперь можно редактировать текст из файла `blinky.c`. Выберите `blinky.c` из окна **Project**. Нажмите его правой кнопкой мыши и выберите **View source file**, или просто дважды щёлкните мышью для того, чтобы просмотреть файл в окне редактирования.

7.3. Работа с отладчиком C51

Обратите внимание, что в режиме отладки на экране видны ещё два окна. Первое – окно кода (рис. 20), где в пошаговом режиме параллельно с исходным текстом на языке C идёт трассировка текста на ассемблере.

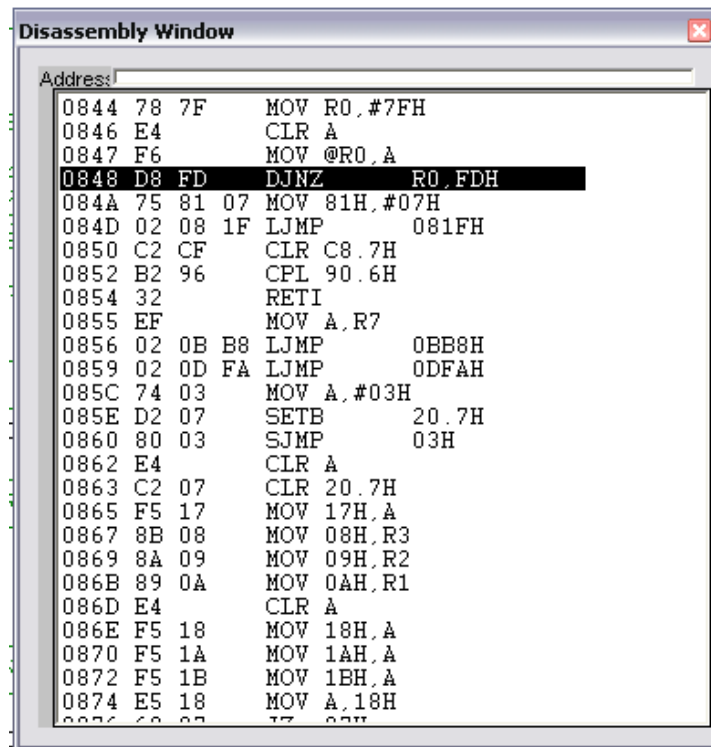


Рис.20. *Окно кода дисассемблера*

Прокрутите окно кода и изучите ассемблерный аналог исходного текста. Обратите внимание на то, сколько кода пришлось бы написать, если проектировать программу на ассемблере.

Ассемблерный аналог текста сохраняется в файле `blinku.lst`. Изучите и постарайтесь понять содержание разделов файла листинга.

Еще одно важное окно, которое должно присутствовать на экране во время отладки, – **View>Debug Window>SFR > 8051 Controller/Misc** (рис. 21).

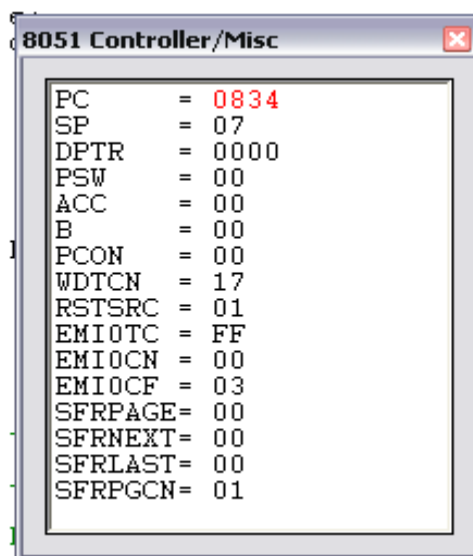


Рис. 21. *Окно основных регистров*

В этом окне постоянно отображается текущее состояние программно-доступных регистров микроконтроллера. Более того, содержимое регистров можно менять во время отладки. Расширенный набор регистров также можно вызывать дополнительными пунктами меню **View>Debug Window>SFR**.

С помощью из меню **View>Debug Window>Memory** можно посмотреть содержимое памяти различного типа в режиме отладки. Попробуйте это сделать.

8. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Перечислите характерные черты архитектуры однокристальных микроконтроллеров.
2. Укажите программно-доступные узлы Intel 8051 и назначение регистров специальных функций.
3. Назовите и охарактеризуйте четыре типа информационных объектов, с которыми может оперировать арифметико-логическое устройство микроконтроллера.
4. Каковы ёмкости памяти программ и памяти данных базового МК Intel 8051 и у МК SiLabs 8051F064?
5. Какой регистр выполняет функции базового регистра при косвенных переходах в программе?
6. Какие операции могут быть выполнены только с использованием аккумулятора?
7. Какие операции могут быть выполнены без участия аккумулятора?
8. Какой формат имеет слово состояния программы Intel 8051? Укажите назначение флагов.
9. Какие возможности предоставляет наличие нескольких банков регистров общего назначения?
10. Как переключить банк регистров общего назначения?
11. Какой регистр используется для адресации внешней памяти данных?
12. Как совместить адресные пространства памяти программ и данных?
13. Охарактеризуйте способ адресации элементов стека в микроконтроллере.
14. Какова длительность исполнения команд в микроконтроллере?
15. Охарактеризуйте режимы работы таймера/счётчика в Intel 8051.
16. Как с помощью таймера можно измерить длительность импульса?
17. Как выводится адрес внешней памяти?
18. Какова нагрузочная способность портов?
19. Перечислите альтернативные функции портов.
20. Охарактеризуйте режимы работы последовательного порта в Intel 8051.
21. Как изменить скорость передачи данных через последовательный порт?
22. Для чего используется девятый бит при передаче данных через последовательный порт?
23. Нарисуйте схему прерываний в Intel 8051. Перечислите и охарактеризуйте типы прерываний.

24. Для чего нужен регистр масок прерывания? Как изменить приоритеты прерываний?
25. Как переводится микроконтроллер в режим пониженного энергопотребления?
26. Охарактеризуйте режим загрузки и верификации программ.
27. Перечислите этапы технологии разработки программ для микроконтроллеров.
28. Укажите назначение основных модулей IDE SiLabs.
29. Что такое объектный код, какие функции выполняет компоновщик?
30. Укажите основные тенденции развития микроконтроллеров.

9. ЗАДАНИЕ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Для углублённого изучения возможностей IDE Silabs и её компонентов самостоятельно изучите содержание и смысл всех пунктов меню, кнопок инструментальной панели, окон и настроек. Для этого воспользуйтесь встроенной справочной системой, которая вызывается через меню **Help**. Эти знания потребуются при выполнении следующих лабораторных работ.

10. СОДЕРЖАНИЕ ОТЧЁТА

Отчёт о лабораторной работе должен содержать:

- титульный лист;
- цель и задачи работы;
- структурную схему микроконтроллера;
- текст программы на ассемблере с комментированием каждой команды;
- текст программы на языке C;
- выводы по работе.

СПИСОК ЛИТЕРАТУРЫ

1. Ваша первая программа для микроконтроллера Intel 8051: Методические указания к лабораторной работе №1 по курсу “Цифровые устройства и микропроцессоры”/ В.А.Добряк. Екатеринбург: Изд-во УГТУ, 1999. 48 с.
2. Горюнов Ф.Г., Ливенцов С.Н. Архитектура микроконтроллера Intel 8051: учебное пособие. Томск: Изд-во ТПУ 2005. 86 с.
3. Магда Ю.С. Микроконтроллеры серии 8051: практический подход. - Издательство: "ДМК Пресс", 2010. – 228 с. URL: http://e.lanbook.com/books/element.php?pl1_id=871
4. Фрунзе А.В. Микроконтроллеры? Это же просто! Т. 1. - Издательство: "Додэка-XXI", 2010. – 311 с. ISBN: 978-5-94120-140-2 URL: http://e.lanbook.com/books/element.php?pl1_id=40964
5. Фрунзе А.В. Микроконтроллеры? Это же просто! Т.2. М.: ООО «Ид Скимен», 2002. 392 с.
6. Фрунзе А.В. Микроконтроллеры? Это же просто! Т.3. М.: ООО «Ид Скимен», 2003. 224 с.
7. Николайчук О. x51 совместимые микроконтроллеры фирмы SYGNAL. М.: ООО «Ид Скимен», 2002. 472 с.
8. Ламберт Е. Как работать с АЦП и ЦАП в микроконтроллерах SiLabs/ Компоненты и технологии, №7, 2005.
9. Гладштейн М.А. Микроконтроллеры смешанного сигнала C8051Fxxx фирмы Silicon Laboratories и их применение. Руководство пользователя. - Издательство: "Додэка-XXI", 2010. – 336 с. URL: http://e.lanbook.com/books/element.php?pl1_id=40973
10. C8051F060-67 Mixed Signal MCU Family.pdf. [Электронный ресурс] <http://www.silabs.com>
11. C8051F064 Evaluation kit user's guide.pdf. [Электронный ресурс] <http://www.silabs.com>